U.S. DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

Nonlinear least-squares inversion
of transient soundings for a
central induction loop system
(Program NLSTCI)

by

Walter L. Anderson

Open-File Report 82-1129

1982

DISCLAIMER

This program was written in FORTRAN-77 for a VAX-11/780 (VMS version 2.5) system*. Although program tests have been made, no guarantee (expressed or implied) is made by the author regarding program correctness, accuracy, or proper execution on all computer systems.

----

* Any use of trade names in this report is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey. This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards.

CONTENTS

INTRODUCTION

The inversion of transient soundings for a central induction loop system on a layered halfspace is provided by program NLSTCI. The numerical technique uses a general adaptive nonlinear least-squares algorithm originally developed by Dennis and others (1979), and extended externally for constrained nonlinear regression by Anderson (1982). The corresponding forward problem solution—also required in the inverse solution—is defined in Anderson (1981). The numerical integrations used in NLSTCI are by adaptive digital linear filtering as described in Anderson (1975) and Anderson (1979). Because digital convolution (filtering) methods are used, practical solutions for layered earth models are reasonably fast on most computers.

This report summarizes the general nonlinear least-squares (NLS) method used in Anderson (1982), but as applied to observed transient soundings obtained using a central induction loop system placed on an assumed horizontally layered earth model. In addition, the quasi-static case is assumed (i.e., displacement currents are neglected). The system must use an "on-off" step current source of arbitrary current, where the transient decay voltage is measured during the off-time (i.e., after t>0 sec.). An arbitrary maximum of 10-layers (homogeneous and isotropic) may be used; however, with most present time-domain electromagnetic (TDEM) measurement systems, only a few layers are generally resolvable for the given time range.

To avoid repeating the notation and other details of the
forward problem solution in this report, the reader is
referred to Anderson (1981)--which has been updated from the
original published version. Similarly, details on the NLS
method may be found in Anderson (1982). The present report
will provide a brief description of the calculations,
specific program parameters, and the VAX operating
instructions. Appendix 1 offers some suggestions in
converting the VAX program to other computer systems;
Appendix 2 lists a simple input/output test example (taken
from a known forward solution model); and Appendix 3 gives
a partial source listing (the complete source is available
on the distributed tape, as described in Appendix 3).

SUMMARY OF CALCULATIONS

The NLS method described in Anderson (1982) requires a
twice-continuously differentiable nonlinear objective
function F describing the model equation as a function of
the unknown layer parameters (i.e., the conductivities and
thicknesses of an MM-layered earth, MM>0). In this case, F
is given by the transient V(t) defined in Anderson (1981,
p.5), as

$$V(t) = \frac{2}{\pi} C \int_0^\infty Re[Hz(\sqrt{b})/DC] \cos(bT) \, db, \qquad (1)$$

where discrete observed values $[V(t_i), t_i, \; i=1,2,\ldots,N]$ are
given. In some cases (e.g., data stacking), an associated
standard deviation s_i may also be known, and should be used

for a weighted least-squares solution (see parameter IWT=1 in Anderson, 1982, p.14). Note that the constant C in eq.(1), and in Anderson (1981, p. 4), should be corrected to read: $C=(nA)I/[\sigma_i(a^2+z^2)^{3/2}]$ --see Anderson (1981, p. 4-5) for definitions of all symbols used.

Optionally, F .may be given in terms of converted apparent resistivity ρ_a (see $INIT parameter IOPT=1 below) instead of voltage V(t) by equating the layered earth transient response to a halfspace response. In this case, the observed transient data $[V(t_i),t_i]$ must be converted to apparent resistivity data $[\rho_a(t),t_i]$ by solving the following nonlinear equation (Raab and Frischknecht, 1982) for x at each i,

$$3\ erf(x)-(3x+2x^3)\frac{2}{\sqrt{\pi}}\exp(-x^2)-2x^2 T_i V(t_i) = 0, \qquad (2)$$

where C=1 is assumed in eq.(1), $T_i=2t_i/(\mu_0\sigma_i a^2)$ is normalized time, and the units are V(volts/amp), t(seconds), and ρ_a(ohm-meters). For each solution of root x at an observed t_i, the apparent resistivity is given by

$$\rho_a = (\mu_0 a^2)/(4t_i x^2). \qquad (3)$$

When F is defined as in eq. (1) and IOPT=0 (default), then any convenient unit may be used for V (e.g., volts/amp, millivolts/amp, etc.), since the constant C in eq. (1) can be determined in the least-squares to account for a scale (or amplitude shift) factor times V(t).

For either IOPT=0 or 1 cases, the independent time variable t>0 must be given in <u>seconds</u> and in ascending order, and is assumed to be known without error.

The unknown (nonlinear) model parameters are denoted by the vector B(J), and has the following assumed order:

B(1),B(2),...,B(MM) are the MM-layer conductivities (in mhos/m.),

B(MM+1),B(MM+2),...,B(2*MM-1) are the MM-1 layer thicknesses (in m.), and

B(2*MM) is a transient scaling (or amplitude shift) parameter depending on the form of F chosen via $INIT parameter IOPT.

Thus, the discrete objective function F may be expressed for either IOPT=0 as

$$F = B(2*MM) \ [V(t_i,B(J), \ J=1,2,...,2*MM-1)/B(1)], \left.\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right\}$$

or for IOPT=1 as

$$F = B(2*MM) \ [\rho_a(t_i,B(J), \ J=1,2,...,2*MM-1)], \qquad (4)$$

where i=1,2,...,N and N>2*MM≥2 (1≤MM≤10). Note that the IOPT=0 form of F has been normalized by the unknown B(1), so that B(2*MM) is a scaling constant free from B(1); the exact form of B(2*MM) can be determined, if desired, and is related to the constant C in eq. (1) above.

In terms of the NLS notation (Anderson, 1982, p.11-12), let $X(I,1)=t_i$ and Y(I) be the observed F in eq. (4), then the observed data matrix is

$$(Y(I),X(I,1),I=1,2,\ldots,N).$$

Since $V(t)$ can range several decades in magnitude for $t_1 \le t \le t_N$, it is advised when IOPT=0 that a weighted least-squares option be used (see IWT=1 or 2, Anderson, 1982, p.14-15), which requires the augmented data matrix

$$(Y(I),X(I,1),X(I,2),I=1,2,\ldots,N),$$

where $X(I,2)=s_i$ is the standard deviation (IWT=1) of observation $Y(I)$, or $X(I,2)$ is the variance (IWT=2). Note that if s_i is unknown, one could use the statistical weight (Bevington, 1969, p.108) of $1/Y(I)$ by setting $X(I,2)=Y(I)$ and IWT=2; in this case, this would be preferred over using unity weights (IWT=0). However when IOPT=1, IWT=0 can generally be used since the range of $\rho_a(t)$ is usually much less than the range of $V(t)$ in most cases.

The analytical partial derivative subprogram (PCODE) was not included in program NLSTCI, therefore the estimated derivative option (IDER=1) must be used, which requires only the forward problem solution subprogram (FCODE). See Appendix 3 listing of FCODE for the coding details, which follows the method described in Anderson (1981) for computing $V(t)$, and as revised for computing $\rho_a(t)$.

Because realizable layered earth models are sought to fit the given data, a constrained minimization type (SP=3 or 4) is advised, along with reasonable lower and higher parameter bound arrays, BL(J) and BH(J) respectively, where

$BL(J) \leq B(J) \leq BH(J)$, J=1,2,...,2*MM (see Anderson, 1982, p.17).
This approach limits parameter space searching, and in some
cases may avoid false starts (or catastrophic overflow
conditions from poor estimates and data). In addition,
individual parameters can be fixed in the least-squares
using parameters IP and IB (Anderson, 1982, p.13). In
particular for the IOPT=1 case, one can usually fix
B(2*MM)=1, provided the observed (converted) apparent
resistivities are properly scaled. Similarly, for the
IOPT=0 case, B(2*MM) can be fixed if the constant C in
eq.(1) is known a priori. [Actually, if the system
calibration is known, then the constant C can be determined;
therefore B(2*MM) should be fixed to reduce the number of
unknowns, and to reduce the possibility of finding an
equivalent but highly improbable solution.] In any case, the
user should attempt to give a reasonable starting guess
vector B(J) corresponding to the given data matrix. It is
advisable to begin with a few layers (e.g., MM=1 or 2)
before trying models with more layers. For present TDEM
equipment, generally only a few layers are all that can be
resolved, due mainly to the small discrete time range
$t_I \leq t \leq t_N$ and noise level in observing V(t).

In general, one should not expect both IOPT=0 and 1 to
yield the same exact solutions for a given data set--due
mainly to data noise, discrete time-range given, scaling,
and the use of different weighting options. For exact data
(as in Appendix 2), both IOPT=0 and 1 produce nearly
identical solution vectors; for noisy observed data, this

is rarely true, although the earth models resolved by both cases should give approximately "equivalent layers" for good fitting cases (i.e., if small parameter errors and RMS error).

PARAMETERS, FILES AND DATA REQUIRED

All $PARMS parameters (excluding the ISTOP=0 option), program files (FOR005-FOR016), and data ordering requirements used by NLSTCI are identical to those described in detail for subprogram NLSOL (Anderson, 1982, p.9-21), and therefore will not be repeated here. However, note that the ordering of the $PARMS estimated parameter vector B(J) used by NLSTCI must be given exactly as described above in eq. (4). The $INIT model parameters required by NLSTCI must be given after the object-time format statement on FOR005 (see Anderson, 1982, p.10, item 5). Also see the EXAMPLE below and Appendix 2 for a typical data input.

$INIT PARAMETER DEFINITIONS

$INIT parameters (nondefault parameters must be given):

MM=   Number of layers in the model ($1 \leq MM \leq 10$; default MM=1 for a homogeneous half-space). Since NLSOL also requires the total number of parameters K, then make sure that K=2*MM is given in $PARMS also. (See the section ERROR MESSAGES below for a discussion on K=2*MM dual input requirement.)

IOPT=0 (default)   means   that   the   data   matrix

(Y(I),X(I,1),I=1,N) is given with Y(I)=V(t) transient data, which may be unscaled and in any units as determined by B(2*MM) in the least-squares solution. X(I,1)=t_i must be given in <u>seconds</u> and in ascending order for I=1,2,...,N.

IOPT=1   means the data matrix (Y(I),X(I,1),I=1,N) is given with Y(I)=ρ_a(t) apparent resistivity data (in ohm-m.). The shift parameter B(2*MM)=1 can be fixed via $PARMS IP,IB provided the apparent resistivity is known to be scaled correctly. X(I,1)=t_i must be given in <u>seconds</u> and in ascending order for I=1,2,...,N. When IOPT=1 is selected, then ISTEP=0 (see below) can only be used.

A=   Radius (in m.) of transmitter circular loop, where A>0 must be given. [Note that a square loop of side L (m.) is considered equivalent to a circular loop of radius A (m.), where A=L/$\sqrt{\pi}$.]

Z=   Tranmitter loop elevation (in m.) on or above the surface (default Z=0.0). Note that Z>0 specifies the source loop is Z meters above the surface, but that the central induction receiver coil is assumed to be placed on the surface. For most field applications, Z=0 is always used. [Z is included here only to maintain compatibility with the forward program solution (Anderson, 1981).]

ISTEP=0 (default) to compute the transient derivative response (TDR) sounding (Anderson,1981), which corresponds to the time-derivative of Hz when the

source uses a system step driving current (e.g., when using a vertical-axis coil at the loop projected center).

ISTEP=1   to compute the transient field response (TFR) sounding (Anderson, 1981), which corresponds to the integral over time of the TDR-sounding. The TFR-sounding (ISTEP=1) is generally used when transient (stacked) data is obtained using a SQUID or cryogenic magnetometer. Note that Z=0 must be used whenever ISTEP=1.

EPS=   Requested convolution integration tolerance used to compute all Fourier and Hankel transforms by digital filtering (default EPS=0.1E-9).

BO=.01   (default) is the lower induction number for which the normalized Hz/DC frequency response approaches the limit 1.0 for B<BO. This assumption saves time by avoiding explicit response calculations for B<BO. BO must be given (or assumed .01 by default) as a power of 10^{**-n} (n integer). The default value is usually adequate for most models; for more accuracy in the late-time transient, BO<.01 can be used. [For accuracy reasons, BO>.01 should never be used.]

BM=100   (default) is the upper induction number for which the normalized Hz/DC frequency response approaches the limit 0.0 for B>BM. This assumption saves time by avoiding explicit response calculations for B>BM. BM must be given (or assumed 100 by default)

as a power of 10**n (n integer). The default value is usually quite adequate for most models; for more accuracy in the early-time transient, BM>100 can be used.

NB=8   (default) represents the number of induction number points per decade (log-cycle) to evaluate the pre-splined frequency response function Hz(B)/DC. In general, $3 \leq NB \leq 11$ is usually adequate for most applications (NB<3 is not recommended for accuracy reasons). If NB=0 (or NB>11) is specified, then a direct mode of evaluating the frequency function is used but as controlled by the outer time-integral via lagged convolution (i.e., the cosine filter using subroutine RLAGFO). Note that NB=0 (or NB>11) is more accurate, but much more time-consuming than using NB<12. [See the section COMPUTER TIMING CONSIDERATIONS for a further discussion on the use on NB.]

$END   [end of $INIT parameters; the "END" in $END may be omitted, if desired.]

EXAMPLE OF INPUT PARAMETERS AND DATA ORDERING

```
EXAMPLE TITLE WITH OBJECT DATA ON FOR005 (IALT=5)
$PARMS N=20,M=1,K=4,IP=1,IB=4,
IDER=1,IPRT=-1, IALT=5,SP=3,IWT=1, NITER=5,
BL=2*.0001,10,.1, B=.1,.01,100,.1,
BH=2*10,1000,.1$
(3F10.0)
0.1        .0004      .18
0.03       .0008      .09
---<etc. for 18 more observations>---
$INIT MM=2,A=100,NB=4,EPS=.1E-5$END
```

(See Appendix 2 for a complete input/output example.)

COMPUTER TIMING CONSIDERATIONS

The computer CPU-time will vary mostly as a function of the given $INIT parameters MM,EPS,BO,BM,NB and $PARMS parameters N,NITER,IP,SP,IV,V, and B. Perhaps the parameters of greatest effect on CPU-time are how good the initial model estimates are given in array B(J), J=1,2,...,2*MM, with respect to the observed data matrix. Of course, the observed data matrix time-range and noise level can contribute further problems in resolving a given layered earth model for any MM in (1,10). In some cases, it may be necessary to fix certain parameters in B (via $PARMS IP,IB) that cannot be resolved and/or to control the initial theoretical transient curve behavior. Generally, it is best to begin with a small MM (say 2 or 3), and progressively increase MM until the RMS error cannot be further decreased. During this "initial model searching study", several $INIT parameters can be modified (relaxed) to significantly reduce the overall CPU-time, but with somewhat less accurate results (which may not be needed for initial runs). Some suggestions are provided in Table 1.

Table 1. Recommended $INIT parameters for NLSTCI

$INIT parameter	Default value	Faster CPU; less accurate	Slower CPU; more accurate
EPS	0.1E-9	0.1E-5	0.1E-11
BO	0.01	0.01*	0.001
BM	100	10	1000
NB	8	2<NB<8	8<NB<12

* BO>0.01 should never be used
   (see Anderson, 1981, p. 25-41).

For a _final_ model run, the _default_ values in Table 1 are generally sufficient for most field situations, with the exception that NB>8 may be used to reduce any noticeable nonsmoothness in the calculated transient. (Note that NB>11 is _not_ recommended for routine field work.)

Some $PARMS parameters used in the NLS algorithm can also be modified to reduce the total CPU-time when searching for an initial model. In particular, $PARMS NITER (Anderson, 1982, p. 16) can be set small (e.g., 3 or 5) to force termination of a trial run after just a few iterations. This is reasonable, since it may not be necessary to obtain normal convergence of the iteration process for preliminary or intermediate models. Other $PARMS that control the NLS algorithm speed and accuracy can also be overridden from their default values (see Table 2 in Anderson, 1982, p. 20-21 for more details).

DATA MATRIX NOTES

The data matrix (defined following eq. (4)) is read under the object-time format statement, and is defined as the sequence of ordered rows:

$$(Y(I),(X(I,L),L=1,M*),I=1,N),$$

where M*=M if IWT=0 (default), or M*=M+1 if IWT=1 or 2. In the above example, IWT=1, M=1, and therefore three columns are required in the data matrix row, where in this case, the last column represents the standard deviation of observation Y(I).

SPECIAL OBJECT FORMAT PHRASES

If an existing data matrix file does not have the proper defined column ordering in the form $(Y(I),X(I,J),J=1,M)$, then the FORTRAN "Tn" format phrase may be used to begin at any column n in the data record. For example, the format $(T41,F10.0,T1,2F10.0)$ will select $Y(I)$ using column 41-50 and $X(I,1)$ beginning at column 1. See any FORTRAN-77 coding manual for other allowable object (run) time format phrases (e.g., the G-format, use of "/" to skip records, etc.). Note that "tab"-characters must <u>not</u> be used when creating the data matrix file FOR010.

VAX OPERATING INSTRUCTIONS

In general, the basic steps described to run NLSOL (Anderson, 1982, p.22-24) can be followed to run NLSTCI either on-line or in batch mode. That is, the parameter and data matrix files may be associated with the logical names FOR005 and FOR010, respectively, using the VAX-DCL statements:

        $ASSIGN parameterfilename FOR005

        $ASSIGN datamatrixfilename FOR010

        $RUN NLSTCI !(use $RUN [WANDERSON]NLSTCI on USGS VAX)

If the data matrix is included on FOR005 (i.e., using IALT=5), then the FOR010 assignment is not necessary.

In addition, program NLSTCI has a useful "restart file" (called FOR005.TMP) that is automatically provided each time the program is executed. File FOR005.TMP contains a copy of all parameters on FOR005, plus the last solution B-vector obtained; note that $PARMS ISTOP=0 (Anderson, 1982, p.14) cannot be used because FOR005 is positioned at EOF in creating FOR005.TMP. If desired, one can easily continue (or restart) more iterations simply by using the DCL commands:

        $ASSIGN FOR005.TMP FOR005

        $RUN NLSTCI !(use $RUN [WANDERSON]NLSTCI on USGS VAX)

Note that FOR005.TMP may also be edited (using any VAX editor) for other parameter changes, if desired. Also, the reassignment of FOR005 using FOR005.TMP only needs to be done once for multiple continuation runs.

By default, the master print (disk) file is called FOR016.DAT, unless otherwise assigned. This file can be TYPEd or PRINTed on a line printer. Also, file FOR016 may be used as an input file to a plot routine; e.g., to plot the observed (OBS), calculated (CAL), and residual (RES) curves. If program NLSTCI is run on-line, then a shorter terminal print file on FOR006 contains some of the information as on FOR016, but as controlled by parameter IPRT (Anderson, 1982, p.15).

ERROR MESSAGES

Almost all $PARMS syntactical errors are flagged and printed on files FOR006 and FOR016 and the job is aborted (see Anderson, 1982, p.24). However, some cross-references (or dual inputs) are not checked; for example, the relationship K=2*MM is not double checked between $PARMS K and $INIT MM parameters. This is because a general-purpose nonlinear least-squares algorithm (NLSOL) is being used as a control program, but the model input is external to the particular nonlinear problem requirements (NLSTCI) read by subprogram SUBZ (see Anderson, 1982, p.38). Therefore, the user is responsible for providing exactly K parameter estimates in B(I),I=1,2,...,K (see eq. (4)), and that $INIT MM is such that K=2*MM (otherwise, unpredictable results could occur).

The message "{WARN}: NOISE IN CALC. TRANS DETECTED" can occur for certain model estimates in array B with respect to the given data matrix. This warning message actually means that the calculated transient voltage V/I cannot be computed accurately at late times using single-precision arithmetic (regardless of the values specified in $INIT parameters EPS,BO,BM, and NB). However, this condition is usually unimportant if the warning occurs near the beginning of the NLS iteration. For typical field data cases, and a moderate MM value and reasonable B estimates, one should not expect the warning message to appear near the end of the NLS iterations for a converging model solution.

PRINTED OUTPUT

All input parameters are output on files FOR006 and FOR016, with the $INIT parameters given first, followed by all $PARMS parameters given or assumed by default. (Refer to Appendix 2 for a complete sample output listing.)

Specific names (e.g., IT, NF, ...) used by NLSOL in the output listings are tabulated in Anderson (1982, p.25-26). Program NLSTCI provides a summary listing of the final solution vector B, along with accumulated layer thicknesses listed under the DEPTH column (see the end of the listing example in Appendix 2). The RESISTIVITY column is simply 1/SIGMA, where SIGMA is the layer conductivity (in mhos/m.).

REFERENCES

Anderson, W.L., 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: USGS Rept. GD-75-012, 223 p. (also available as NTIS Rept. PB-242-800.)

————————, 1979, Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics, v. 44, n. 7, p. 1287-1305.

————————, 1981, Calculation of transient soundings for a central induction loop system (Program TCILOOP): USGS Open-File Rept. 81-1309, 80 p.

————————, 1982, Adaptive nonlinear least-squares solution for constrained or unconstrained minimization problems

(Subprogram NLSOL): USGS Open-File Rept. 82-68, 65 p.

Bevington, P.R., 1969, Data reduction and error analysis for
the physical sciences: McGraw-Hill, N.Y., 336 p.

Dennis, J.E., Gay, D.M., and Welsch R.E., 1979, An adaptive
nonlinear least-squares algorithm: Univ. of Wisconsin
MRC Tech. Sum. Rept. 2010 (also available as NTIS
Rept. AD-A079-716),

Raab, P.V., and Frischknecht, F.C., 1982, Desk top programs
for central and coincident loop TDEM calculations:
USGS Open-File Rept. 82-xxx, yy p [in press].

Appendix 1.-- Conversion to other systems

This program (and associated subprograms) was written in ANSI-standard FORTRAN-77 for the VAX-11/780 system (VMS version 2.5). Conversion to systems without an ANSI-FORTRAN-77 compiler would necessitate extensive changes, particularly for all CHARACTER-type variables, IF-THEN-ELSE phrases, etc.

Since the FORTRAN-77 ANSI-standard presently does not provide for a NAMELIST I/O capability, a VAX-11 NAMELIST simulator subprogram is included in this program package. For most large main-frame systems (e.g., IBM/370, CYBER, etc.), a NAMELIST READ/WRITE is usually available; in this case, the VAX NAMELIST subprogram and associated routines (DECODEIX, DECODEX) can be eliminated; also, appropriate changes can be made where COMMON/NAME_LIST/ and CALL NAMELIST is used in the source program.

Other changes for non-VAX systems might include some (or all) of the following:

(1) Variables with more than 6-characters.

(2) Use of the underscore character or dollar character in some variables and/or COMMON names.

(3) Character strings delimited by single-quote characters (e.g., 'STRING'); also, character string concatination (e.g., 'STRING1'//'STRING2').

(4) Passing variable-length character strings in subroutine calls; e.g., CHARACTER*(*) passed length character

arguments.

(5) Need to suppress arithmetic or exponential underflow
messages (note that a VAX-11 result is automatically set
to 0.0 after any underflow--which is assumed for this
program package); if the target system does not set
underflows to 0.0 (and suppress warning messages), then
a suitable conversion procedure must be used for proper
operation of this program package.

(6) Replacement of any special VAX-dependent CALLS or
statements (e.g., CALL LIB$INDEX, ACCEPT, TYPE, CALL
SYS$anyname, etc.--note that we have minimized
machine-dependent calls, where possible).

(7) Hexidecimal constants (e.g., '4A'X) if used in any DATA
statements.

(8) Virtual-sized arrays, if any (i.e, DIMENSION statements
greater than physical memory).

Appendix 2.-- Test problem input/output listing

The following input files (FOR005.0, FOR010, FOR005.1) were used to run a known test problem for program NLSTCI on a VAX system using both IOPT=0 and 1 cases separately. The corresponding output files (FOR016) are given following FOR005.1. In addition, each file FOR016.DAT was used to plot the final observed (OBS) and calculated (CAL) curves using an external plotter. The symbol "0" represents Y(I) in the plot, and the solid line represents a curve drawn through the calculated (CAL) points.

FOR005.0

```
TEST EXAMPLE (IOPT=0 CASE)
$PARMS N=19,K=4,M=1,IPRT=-2,
IDER=1,IWT=2,SP=3,
NITER=15,
BL=2*.0001,10,.1E-4,
B=.015,.15,175,.015,
BH=2*5,1000,.1E5$
(2G16.8,T1,G16.8)
$INIT MM=2,A=175$
```

FOR010

```
0.96605726E-01      0.19242254E-03      0.11366887E+03
0.36346640E-01      0.28243766E-03      0.12427132E+03
0.13687826E-01      0.41456183E-03      0.13160466E+03
0.64035309E-02      0.60849357E-03      0.11748647E+03
0.35103841E-02      0.89314644E-03      0.93367012E+02
0.21471761E-02      0.13109597E-02      0.68623184E+02
0.13093358E-02      0.19242257E-02      0.50529686E+02
0.78840711E-03      0.28243773E-02      0.37524563E+02
0.45996808E-03      0.41456190E-02      0.28474813E+02
0.25708214E-03      0.60849362E-02      0.22246769E+02
0.13831072E-03      0.89314654E-02      0.17830414E+02
0.71406452E-04      0.13109598E-01      0.14687531E+02
0.35465542E-04      0.19242259E-01      0.12410449E+02
0.16997505E-04     ·0.28243775E-01      0.10733011E+02
0.78670519E-05      0.41456193E-01      0.94950829E+01
0.35501544E-05      0.60849369E-01      0.85372982E+01
0.15584020E-05      0.89314662E-01      0.78144689E+01
0.66868074E-06      0.13109601E+00      0.72586303E+01
0.28317137E-06      0.19242261E+00      0.67987070E+01
```

FOR005.1

```
TEST EXAMPLE (IOPT=1 CASE)
$PARMS N=19,K=4,M=1,IPRT=-2,
IDER=1,IWT=0,SP=3,
NITER=15,IP=1,IB=4,
BL=2*.0001,10,.1E-4,
B=.015,.15,175,1,
BH=2*5,1000,.1E5$
(T33,G16.8,T17,G16.8)
$INIT MM=2,A=175,IOPT=1$
```

FOR016


```
(NLSTCI):      TEST EXAMPLE (IOPT=0 CASE)

MM=  2              A= 0.175000E+03    EPS= 0.100000E-09
BO= 0.100000E-01   BM= 0.100000E+03   NB=  8
Z= 0.000000E+00    ISTEP=  0
IOPT=  0



PARAMETER ORDER--

        1        SIGMA(  1)
        2        SIGMA(  2)
        3        THICK(  1)
        4           B(  4) SHIFT PARAMETER IN B(2*MM)*TRANSIENT
```

(NLSOL):   TEST EXAMPLE (IOPT=0 CASE)

N=	19	K=	4	IP=	0	M=	1	IALT=	10
ISTOP=	1	IWT=	2	IDER=	1	IPRT=	-2	NITER=	15
IOUT=	1	SP=	3						

FMT=(2G16.8,T1,G16.8)

PARAMETER LOWER BOUNDS: BL=

0.99999997E-04   0.99999997E-04   0.10000000E+02   0.99999997E-05

INITIAL PARAMETERS: B=

0.15000000E-01   0.15000001E+00   0.17500000E+03   0.15000000E-01

PARAMETER HIGHER BOUNDS: BH=

0.50000000E+01   0.50000000E+01   0.10000000E+04   0.10000000E+05

** NLITR (IDER=0) OR NL2SNO (IDER=1) CALLED: 1 **

I	INITIAL X(I)	D(I)
1	0.546171E-01	0.490E+02
2	0.174026E+00	0.106E+01
3	0.420534E+00	0.230E+01
4	0.122434E-02	0.186E+04

IT	NF	F	DF	COSMAX	VAR
0	1	0.200E+00		0.999E+00	
1	2	0.110E-01	0.189E+00	0.994E+00	0.150E+02
2	3	0.831E-04	0.109E-01	0.924E+00	0.150E+02
3	4	0.756E-05	0.755E-04	0.946E+00	0.149E+02
4	5	0.324E-06	0.724E-05	0.655E+00	0.145E+02
5	6	0.118E-06	0.206E-06	0.656E+00	0.140E+02
6	7	0.225E-07	0.959E-07	0.686E+00	0.123E+02
7	8	0.909E-08	0.135E-07	0.911E+00	0.141E+02
8	9	0.650E-10	0.903E-08	0.995E-01	0.151E+02
9	10	0.636E-10	0.138E-11	0.186E+00	0.144E+01
10	11	0.433E-10	0.202E-10	0.439E+00	0.146E+01
11	12	0.433E-10	-0.117E-10	0.439E+00	0.818E+01

***** X-CONVERGENCE *****

FUNCTION	0.433499D-10	VARIABILITY	0.818060E+01
FUNC. EVALS	12	GRAD. EVALS	11
GRAD. NORM	0.104053E-02	COSMAX	0.438530E+00

I	FINAL X(I)	D(I)	G(I)
1	0.445184E-01	0.430E+02	0.404E-04
2	0.201355E+00	0.426E+00	0.151E-05
3	0.453441E+00	0.162E+01	0.504E-05
4	0.999343E-03	0.131E+04	0.104E-02

COVARIANCE = SCALE * (J**T * J)**-1

ROW 1   0.3025E-11

FOR005.1

```
TEST EXAMPLE (IOPT=1 CASE)
$PARMS N=19,K=4,M=1,IPRT=-2,
IDER=1,IWT=0,SP=3,
NITER=15,IP=1,IB=4,
BL=2*.0001,10,.1E-4,
B=.015,.15,175,1,
BH=2*5,1000,.1E5$
(T33,G16.8,T17,G16.8)
$INIT MM=2,A=175,IOPT=1$
```

FOR016

```
{NLSTCI}:      TEST EXAMPLE (IOPT=0 CASE)

MM=  2              A= 0.175000E+03    EPS= 0.100000E-09
BO= 0.100000E-01  BM= 0.100000E+03    NB=  8
Z= 0.000000E+00   ISTEP=  0
IOPT=  0
```

```
PARAMETER ORDER--

        1       SIGMA(  1)
        2       SIGMA(  2)
        3       THICK(  1)
        4           B(  4) SHIFT PARAMETER IN B(2*MM)*TRANSIENT
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=          282.81 SEC. (    4 MIN. 42.81 SEC.)
CPU TIME=          263.12 SEC. (    4 M. 23.12 S.)    CPU % = 93.04%
BUF.I/O_COUNT=          7
DIR.I/O_COUNT=          20
PAGE_FAULTS=          148
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

TEST EXAMPLE (IOPT=0 CASE)

TRANSIENT

TIME (SEC.)

```
ROW  2    0.2246E-10  0.2355E-09
ROW  3   -0.1491E-10 -0.1127E-09  0.7896E-10
ROW  4   -0.7822E-13 -0.5524E-12  0.3802E-12  0.2047E-14
```

I	OBS.Y(I)	CAL	RES	%RES.ERR	X(I,1)	X(I,2)	X(I,3)	X(I,4)	WT(I)
1	0.966057E-01	0.966070E-01	-0.132E-05	-0.136507E-02	0.192423E-03	0.966057E-01	0.000000E+00	0.000000E+00	0.103514E+02
2	0.363466E-01	0.363465E-01	0.156E-06	0.430474E-03	0.282438E-03	0.363466E-01	0.000000E+00	0.000000E+00	0.275129E+02
3	0.136878E-01	0.136875E-01	0.278E-06	0.203444E-02	0.414562E-03	0.136878E-01	0.000000E+00	0.000000E+00	0.730576E+02
4	0.640353E-02	0.640327E-02	0.262E-06	0.408700E-02	0.608494E-03	0.640353E-02	0.000000E+00	0.000000E+00	0.156164E+03
5	0.351038E-02	0.351037E-02	0.158E-07	0.451021E-03	0.893146E-03	0.351038E-02	0.000000E+00	0.000000E+00	0.284869E+03
6	0.214718E-02	0.214698E-02	0.196E-06	0.910943E-02	0.131096E-02	0.214718E-02	0.000000E+00	0.000000E+00	0.465728E+03
7	0.130934E-02	0.130925E-02	0.851E-07	0.649987E-02	0.192423E-02	0.130934E-02	0.000000E+00	0.000000E+00	0.763746E+03
8	0.788407E-03	0.788485E-03	-0.777E-07	-0.985526E-02	0.282438E-02	0.788407E-03	0.000000E+00	0.000000E+00	0.126838E+04
9	0.459968E-03	0.459998E-03	-0.295E-07	-0.642186E-02	0.414562E-02	0.459968E-03	0.000000E+00	0.000000E+00	0.217406E+04
10	0.257082E-03	0.257110E-03	-0.277E-07	-0.107876E-01	0.608494E-02	0.257082E-03	0.000000E+00	0.000000E+00	0.388981E+04
11	0.138311E-03	0.138347E-03	-0.358E-07	-0.259069E-01	0.893147E-02	0.138311E-03	0.000000E+00	0.000000E+00	0.723010E+04
12	0.714065E-04	0.714202E-04	-0.137E-07	-0.191831E-01	0.131096E-01	0.714065E-04	0.000000E+00	0.000000E+00	0.140043E+05
13	0.354655E-04	0.354660E-04	-0.469E-09	-0.132324E-02	0.192423E-01	0.354655E-04	0.000000E+00	0.000000E+00	0.281964E+05
14	0.169975E-04	0.170017E-04	-0.416E-08	-0.244898E-01	0.282438E-01	0.169975E-04	0.000000E+00	0.000000E+00	0.588322E+05
15	0.786705E-05	0.786947E-05	-0.241E-08	-0.306730E-01	0.414562E-01	0.786705E-05	0.000000E+00	0.000000E+00	0.127112E+06
16	0.355015E-05	0.355112E-05	-0.963E-09	-0.271226E-01	0.608494E-01	0.355015E-05	0.000000E+00	0.000000E+00	0.281678E+06
17	0.155840E-05	0.155997E-05	-0.157E-08	-0.100717E+00	0.893147E-01	0.155840E-05	0.000000E+00	0.000000E+00	0.641683E+06
18	0.668681E-06	0.668978E-06	-0.298E-09	-0.445075E-01	0.131096E+00	0.668681E-06	0.000000E+00	0.000000E+00	0.149548E+07
19	0.283171E-06	0.283069E-06	0.103E-09	0.362866E-01	0.192423E+00	0.283171E-06	0.000000E+00	0.000000E+00	0.353143E+07

** HMSERR= 0.36189712E-06

```
CORRELATION MATRIX
 1  0.1000E+01
 2  0.8414E+00  0.1000E+01
 3 -0.9648E+00 -0.8264E+00  0.1000E+01
 4 -0.9940E+00 -0.7957E+00  0.9458E+00  0.1000E+01
```

```
 **PARM_SOL.   STD_ERROR   REL_ERROR    % ERROR **

 1  0.1000E-01  0.1739E-05  0.1739E-03  0.1739E-01
 2  0.2001E+00  0.1535E-04  0.7670E-04  0.7670E-02
 3  0.2000E+03  0.8886E-05  0.4443E-07  0.4443E-05
 4  0.9997E-02  0.4524E-07  0.4525E-05  0.4525E-03
```

******** E N D ********   TEST EXAMPLE (IOPT=0 CASE)

PARAMETER NAME	FINAL SOLUTION	RESISTIVITY	LAYER DEPTH
1 SIGMA(1) = 0.10002709E-01		1 0.99972916E+02	
2 SIGMA(2) = 0.20009081E+00		2 0.49977307E+01	
3 THICK(1) = 0.19997893E+03			1 0.19997893E+03
4 SHIFT = 0.99968594E-02			

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=          282.81 SEC. (   4 MIN. 42.81 SEC.)
CPU_TIME=          263.12 SEC. (   4 M. 23.12 S.)   CPU % = 93.04%
BUF.I/O_COUNT=        7
DIR.I/O_COUNT=       20
PAGE_FAULTS=        148
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

TEST EXAMPLE (IOPT=0 CASE)

TIME (SEC.)

(NLSTCI):   TEST EXAMPLE (IOPT=1 CASE)

MM= 2   A= 0.175000E+03   EPS= 0.100000E-09
BO= 0.100000E-01 BM= 0.100000E+03   NB= 8
Z= 0.000000E+00   ISTEP= 0
IOPT= 1

PARAMETER ORDER--

        1   SIGMA(1)
        2   SIGMA(2)
        3   THICK(1)
        4   B(4) SHIFT PARAMETER IN B(2*MM)*APPRES

```
{NLSOL}:          TEST EXAMPLE (IOPT=1 CASE)

N=        19    K=        4    IP=        1    M=         1    IALT=   10
ISTOP=     1    IWT=      0    IDER=      1    IPRT=     -2    NITER=  15
IOUT=      1    SP=       3

PARAMETERS HELD FIXED: IB=  4

FMT=(T33,G16.8,T17,G16.8)


PARAMETER LOWER BOUNDS: BL=

  0.99999997E-04   0.99999997E-04   0.10000000E+02   0.99999997E-05

INITIAL PARAMETERS: B=

  0.15000000E-01   0.15000001E+00   0.17500000E+03   0.10000000E+01

PARAMETER HIGHER BOUNDS: BH=

  0.50000000E+01   0.50000000E+01   0.10000000E+04   0.10000000E+05

PARAMETER INDEX:   1   2   3   4
REORDERED AS...:   1   2   3

REORDERED PARAMETERS:

  0.15000000E-01   0.15000001E+00   0.17500000E+03

** NLITR (IDER=0) OR NL2SNO (IDER=1) CALLED:  1 **


    I      INITIAL X(I)       D(I)

    1      0.546171E-01      0.638E+04
    2      0.174026E+00      0.355E+03
    3      0.420534E+00      0.518E+03

   IT   NF       F          DF         COSMAX        VAR

    0    1   0.468E+04                0.984E+00
    1    2   0.186E+04   0.282E+04    0.993E+00    0.159E+02
    2    3   0.478E+02   0.181E+04    0.814E+00    0.159E+02
    3    4   0.420E+00   0.474E+02    0.720E+00    0.157E+02
    4    5   0.234E-02   0.418E+00    0.782E+00    0.150E+02
    5    6   0.383E-04   0.231E-02    0.240E+00    0.155E+02
    6    7   0.383E-04  -0.602E-04    0.240E+00    0.118E+01

***** X-CONVERGENCE *****

FUNCTION      0.383166D-04   VARIABILITY   0.118399E+01
FUNC. EVALS        7         GRAD. EVALS        6
GRAD. NORM    0.720512E+01   COSMAX        0.239794E+00

    I       FINAL X(I)        D(I)           G(I)

    1      0.445123E-01     0.125E+05      -0.697E+01
    2      0.201312E+00     0.393E+03.      0.254E-01
    3      0.453472E+00     0.874E+03       0.183E+01

COVARIANCE = SCALE * (J**T * J)**-1
```

```
ROW  1    0.5643E-13
ROW  2    0.1198E-11  0.5670E-10
ROW  3   -0.1210E-12 -0.2919E-11  0.6546E-11
```

I	OBS.Y(I)	CAL	RES	%RES.ERR	X(I,1)	X(I,2)	X(I,3)	X(I,4)	WT(I)
1	0.113669E+03	0.113669E+03	0.107E-03	0.939673E-04	0.192423E-03	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
2	0.124271E+03	0.124270E+03	0.900E-03	0.724443E-03	0.282438E-03	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
3	0.131605E+03	0.131603E+03	0.204E-02	0.155368E-02	0.414562E-03	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
4	0.117486E+03	0.117492E+03	-0.530E-02	-0.451302E-02	0.608494E-03	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
5	0.933670E+02	0.933657E+02	0.130E-02	0.138916E-02	0.893146E-03	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
6	0.686232E+02	0.686239E+02	-0.717E-03	-0.104506E-02	0.131096E-02	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
7	0.505297E+02	0.505287E+02	0.973E-03	0.192514E-02	0.192423E-02	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
8	0.375246E+02	0.375267E+02	-0.216E-02	-0.574339E-02	0.282438E-02	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
9	0.284748E+02	0.284767E+02	-0.188E-02	-0.659746E-02	0.414562E-02	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
10	0.222468E+02	0.222456E+02	0.117E-02	0.524732E-02	0.608494E-02	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
11	0.178304E+02	0.178293E+02	0.113E-02	0.632242E-02	0.893147E-02	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
12	0.146875E+02	0.146877E+02	-0.154E-03	-0.104538E-02	0.131096E-01	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
13	0.124104E+02	0.124118E+02	-0.135E-02	-0.108877E-01	0.192423E-01	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
14	0.107330E+02	0.107327E+02	0.283E-03	0.263904E-02	0.282438E-01	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
15	0.949508E+01	0.949251E+01	0.258E-02	0.271459E-01	0.414562E-01	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
16	0.853730E+01	0.853363E+01	0.367E-02	0.429921E-01	0.608494E-01	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
17	0.781447E+01	0.781489E+01	-0.420E-03	-0.537555E-02	0.893147E-01	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
18	0.725863E+01	0.726051E+01	-0.188E-02	-0.258827E-01	0.131096E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01
19	0.679871E+01	0.680066E+01	-0.196E-02	-0.287617E-01	0.192423E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.100000E+01

** RMSERR= 0.21885093E-02

CORRELATION MATRIX
```
1   0.1000E+01
2   0.6699E+00  0.1000E+01
3  -0.1991E+00 -0.1515E+00  0.1000E+01
```

**PARM_SOL.	STD_ERROR	REL_ERROR	% ERROR **
1 0.1000E-01	0.2376E-06	0.2376E-04	0.2376E-02
2 0.2000E+00	0.7530E-05	0.3765E-04	0.3765E-02
3 0.2000E+03	0.2559E-05	0.1279E-07	0.1279E-05

******** E N D ********   TEST EXAMPLE (IOPT=1 CASE)

PARAMETER NAME	FINAL SOLUTION		RESISTIVITY	LAYER DEPTH
1 SIGMA(1) =	0.10000006E-01	1	0.99999939E+02	
2 SIGMA(2) =	0.20000516E+00	2	0.49998713E+01	
3 THICK(1) =	0.20000261E+03			1 0.20000261E+03
4 SHIFT =	0.10000000E+01			

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=          184.76 SEC. (   3 MIN.   4.76 SEC.)
CPU_TIME=        167.67 SEC. (   2 M.  47.67 S.)    CPU % = 90.75%
BUF.I/O_COUNT=           7
DIR.I/O_COUNT=          19
PAGE_FAULTS=           153
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

TEST EXAMPLE (IOPT=1 CASE)

APPARENT RESISTIVITY (OHM-M.)

TIME (SEC.)

Appendix 3.-- Source code availability and listing

Source Code Availability

The current version of the source code may be obtained by
writing directly to the author*. A magnetic tape copy can
be sent to requestors to be copied and returned.   This
method of releasing the source code was selected in order to
satisfy requests for the latest (e.g., possibly updated)
version.   (The attached listing does not include the
adaptive nonlinear least-squares algorithm (Dennis   and
others, 1979) due to its length; however, the complete
algorithm is available on the distributed tape.)

The magnetic tape is usually recorded in the following
mode (unless requested otherwise):

Industry compatible: 9-track, standard ANSI-labeled,
ASCII-mode, odd-parity, 800-bpi density, 80-character
card-image records (blocked 50-card images,   or
4000-characters, per physical block), and contained on
a file named "NLSTCI.VAX".

------
* present address is:

U.S. Geological Survey
Mail Stop 964
Box 25046, Denver Federal Center
Denver, CO 80225

Source Listing

The attached subprograms are listed in the following order:

```
00000010   [MAIN PROGRAM]
00000170   REAL FUNCTION HZLOOP
00000470   COMPLEX FUNCTION F3ZH
00000610   SUBROUTINE RECUR
00000840   SUBROUTINE MARQ_TRANS_HZLOOP_FCODE
00002500   REAL*8 FUNCTION FCTCI
00002650   SUBROUTINE MARQ_TRANS_HZLOOP_SUBZ
00003780   SUBROUTINE NAMELIST
00008870   SUBROUTINE DUMYPCODE
00008910   SUBROUTINE SIGSUBEND
00009760   SUBROUTINE CPUTIME
00010330   SUBROUTINE DECODEIX
00010490   SUBROUTINE DECODEX
00010660   REAL*8 FUNCTION DERF
00011080   SUBROUTINE DFIND
00011440   SUBROUTINE DRTMI
00012210   SUBROUTINE ERRMSG
00012550   SUBROUTINE INTEG1
00012800   SUBROUTINE MINMAX
00012900   SUBROUTINE NLSOL
00019190   SUBROUTINE NLITR
00020250   SUBROUTINE INTRAN
00020840   SUBROUTINE CALCR
00021330   SUBROUTINE NONBLANK
00021460   SUBROUTINE PROCINFO
00021830   REAL FUNCTION RFLAGS
00022240   SUBROUTINE SPLIN1
00023440   SUBROUTINE SPOINT
00023660   SUBROUTINE WARN
00024000   COMPLEX FUNCTION ZHANKS
00027440   REAL FUNCTION ASINH
00027520   FUNCTION ERF
00027850   FUNCTION ERFINV
00028650   INTEGER FUNCTION LOC
00028760   SUBROUTINE NL2SOL
00033330   SUBROUTINE NL2SNO
00034880   SUBROUTINE NL2ITR
00041960   SUBROUTINE ASSESS
00045960   SUBROUTINE COVCLC
00050120   SUBROUTINE DFAULT
00051010   REAL FUNCTION DOTPRD
00051380   SUBROUTINE DUPDAT
00051960   SUBROUTINE GQTSTP
00057880   SUBROUTINE ITSMRY
00060180   SUBROUTINE LINVRT
00060610   SUBROUTINE LITVMU
00060930   SUBROUTINE LIVMUL
```

```
                      00061240   SUBROUTINE LMSTEP
                      00066350   SUBROUTINE LSQRT
                      00067000   REAL FUNCTION LSVMIN
                      00068790   SUBROUTINE LTSQAR
                      00069150   SUBROUTINE PARCHK
                      00071070   SUBROUTINE QAPPLY
                      00071970   SUBROUTINE QRFACT
                      00074360   SUBROUTINE RPTMUL
                      00075110   SUBROUTINE SLUPDT
                      00075730   SUBROUTINE SLVMUL
                      00076190   LOGICAL FUNCTION STOPX
                      00076420   SUBROUTINE VAXPY
                      00076550   SUBROUTINE VCOPY
                      00076680   SUBROUTINE VSCOPY
                      00076810   REAL FUNCTION V2NORM
                      00077360   INTEGER FUNCTION IMDCON
                      00077530   REAL FUNCTION RMDCON
                      00078570     REAL FUNCTION RLAGF0
                      00080960     REAL FUNCTION RLAGF1
                      00083320   FUNCTION TCHEB
```

```
C (NLSTCI): 'NLSOL'-INVERSION OF TRANSIENT SOUNDINGS FOR  (8/9/82)      00000010
C  A CENTRAL INDUCTION LOOP SYSTEM OF RADIUS A>0.                        00000020
C                                                                        00000030
C** VAX-11/780 VERSION                                                   00000040
C                                                                        00000050
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.            00000060
C                                                                        00000070
C                                                                        00000080
      EXTERNAL MARQ_TRANS_HZLOOP_FCODE,DUMYPCODE,                        00000090
     1 MARQ_TRANS_HZLOOP_SUBZ,SIGSUBEND                                  00000100
      CALL SETTIME                                                       00000110
      CALL NLSOL(MARQ_TRANS_HZLOOP_FCODE,DUMYPCODE,                      00000120
     1 MARQ_TRANS_HZLOOP_SUBZ,SIGSUBEND)                                 00000130
      CALL CPUTIME(6,16)                                                 00000140
      CALL EXIT                                                          00000150
      END                                                               00000160
      REAL FUNCTION HZLOOP(B2)                                          00000170
C--COSINE-TRANSFORM KERNEL FOR CENTRAL INDUCTION LOOP WITH               00000180
C  A>0,R=0, AND Z>=0.0.                                                  00000190
C                                                                        00000200
      REAL SIG(10),H(10),Z                                              00000210
      COMPLEX ZHANKS,ZAC4,K2(10),KS1,ZFLD                               00000220
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                                00000230
      COMMON/PASS/ZAC4,ANORM,CURI,DC,SIG,B0,BM,SIG1,EPS,ISTEP           00000240
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN      00000250
      EXTERNAL F3ZH                                                     00000260
      B=SQRT(B2)                                                        00000270
      IF(B.LT.B0) GO TO 3                                              00000280
      IF(B.GT.BM) GO TO 4                                              00000290
      IF(ISPLN.EQ.0) GO TO 10                                          00000300
C--ISPLN=1 (0<NB<12 OPTION) INTERPOLATE PRE-SPLINED FREQ. FUNCTION       00000310
      CALL SPOINT(NS,XS,YS,AS,BS,CS,B,HZLOOP)                          00000320
      RETURN                                                            00000330
```

```
10     F=(B/A)**2/(39.47841762E-7*SIG1)                         00000340
       KS1=CMPLX(0.0,-7.895683523E-6*F)                         00000350
       DO 1 I=1,M                                               00000360
1      K2(I)=KS1*CMPLX(SIG(I),0.0)                              00000370
       ZFLD=ANORM*ZHANKS(1,ANORM,F3ZH,EPS,LL,1) + ZAC4         00000380
       ZFLD=CMPLX(CURI,0.0)*ZFLD                                00000390
       HZLOOP=REAL(ZFLD)/DC                                     00000400
       RETURN                                                   00000410
3      HZLOOP=1.0                                               00000420
       RETURN                                                   00000430
4      HZLOOP=0.0                                               00000440
       RETURN                                                   00000450
       END                                                      00000460
       COMPLEX FUNCTION F3ZH(X)                                 00000470
C--KERNEL FOR HANKEL TRANSFORM IN CURLOOP WHEN R=0.0 AND Z>=0.0  00000480
C   SCALED BY HMAX STORED IN COMMON/MODEL/                      00000490
C                                                               00000500
       COMPLEX Z1,Z0,K2(10),KS1,HALF                           00000510
       REAL H(10),Z                                             00000520
       COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                       00000530
       DATA HALF/(0.5,0.0)/                                     00000540
       Y=X/HMAX                                                 00000550
       CALL RECUR(Y,Z1,Z0)                                      00000560
       F3ZH=CMPLX(Y,0.0)*(Z1/(Z0+Z1)-HALF)                     00000570
       IF(Z.GT.0.0) F3ZH=F3ZH*CMPLX(EXP(-Y*Z),0.0)            00000580
       RETURN                                                   00000590
       END                                                      00000600
       SUBROUTINE RECUR(Y,Z1,Z0)                                00000610
C--BACKWARD RECURRENCE FOR COMPLEX IMPEDANCES Z1,Z0 GIVEN ARGUMENT 00000620
C    Y(=X/HMAX) AND MODEL PARAMETERS IN COMMON/MODEL/           00000630
C                                                               00000640
       REAL H(10),Z                                             00000650
       COMPLEX Z1,Z0,K2(10),KS1,ONE,ZZ,X2,U                    00000660
       COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                       00000670
       DATA ONE/(1.0,0.0)/                                      00000680
       X2=CMPLX(Y*Y,0.0)                                        00000690
       Z0=KS1/CMPLX(Y,0.0)                                      00000700
       Z1=KS1/CSQRT(X2-K2(M))                                   00000710
       IF(M.EQ.1) GO TO 20                                      00000720
       J=M-1                                                    00000730
10     U=CSQRT(X2-K2(J))                                        00000740
       ZZ=KS1/U                                                 00000750
       U=CEXP(CMPLX(-2.0*H(J),0.0)*U)                          00000760
       U=(ONE-U)/(ONE+U)                                        00000770
       Z1=ZZ*((Z1+ZZ*U)/(ZZ+Z1*U))                            00000780
       IF(J.EQ.1) GO TO 20                                      00000790
       J=J-1                                                    00000800
       GO TO 10                                                 00000810
20     RETURN                                                   00000820
       END                                                      00000830
       SUBROUTINE MARQ_TRANS_HZLOOP_FCODE(Y,X,B,PRNT,F,IN,IDER) 00000840
C--FUNCT. EVAL. FOR 'NLSTCI'                                    00000850
C                                                               00000860
C--PARAMETERS--                                                 00000870
C      Y=       OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)     00000880
```

```
C        X=       OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)        00000890
C        B=       CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)            00000900
C        PRNT=    WORK AND PRINT ARRAY (DIM. 5)                         00000910
C        F=       OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN 00000920
C        IN=      OBSERVATION NO. TO EVAL. F (1<=IN<=N)                 00000930
C        IDER=    0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED) 00000940
C                 1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED) 00000950
C [NOTE: CURRENTLY ONLY IDER=1 CAN BE USED; IDER=0 MAY BE ADDED LATER]  00000960
C                                                                      00000970
      REAL*8 X0,X1,TV,FX1,SQPI,XL,XR                                    00000980
      PARAMETER (SQPI=1.772453850905516D0)                             00000990
      COMPLEX K2(10),KS1,C4,ZA,ZAC4                                    00001000
      REAL Y(1),X(500,5),B(1),PRNT(5),SIG(10),H(10),DER(2),           00001010
     1 BSAVE(20),W2(200), APPRES(500),TAR(500,2)                      00001020
      EXTERNAL HZLOOP,FCTCI                                            00001030
      COMMON/PASSER/TV,LATE                                            00001040
      COMMON/TCOM/T(500),VSAVE(500)                                    00001050
      COMMON/PASS/ZAC4,ANORM,CURI,DC,SIG,B0,BM,SIG1,EPS,ISTEP         00001060
      COMMON/FPASS/AA,TMIN,TMAX,T0,TM,DB,BMTEST,                       00001070
     * M1,M21,M2,JSPLN,NN,IFIRST,IOPT                                  00001080
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN    00001090
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                               00001100
      DATA DER/2*0.0/,XMU0/1.2566371E-//                               00001110
      IF(IN.GT.1.OR.M.EQ.1) GO TO 20                                   00001120
      DO 10 J=2,M                                                      00001130
      IF(B(J).EQ.B(J-1)) CALL ERRMSG('SOME SIG(J)=SIG(J-1)',4,6,16)   00001140
10    CONTINUE                                                         00001150
20    DO 30 J=1,5                                                      00001160
30    PRNT(J)=X(IN,J)                                                  00001170
      IF(IN.GT.1) GO TO 800                                            00001180
      IF(IDER.EQ.1) GO TO 8001                                         00001190
35    SIG1=B(1)                                                        00001200
      HMAX=A                                                           00001210
      IF(M.EQ.1) GO TO 45                                              00001220
      DO 40 J=1,M1                                                     00001230
      H(J)=B(M+J)                                                      00001240
40    SIG(J)=B(J)                                                      00001250
      CALL MINMAX(H,M1,HMIN,HMAX)                                     00001260
45    SIG(M)=B(M)                                                      00001270
      ANORM=A/HMAX                                                     00001280
      TCON=6.28318531E-7*SIG1*AA                                       00001290
      IF(JSPLN.EQ.0) GO TO 49                                          00001300
C--GET PRE-SPLINED FREQ FUNCTION (0<NB<12 OPTION)                      00001310
      NS=0                                                             00001320
      TEM=B0/DB                                                        00001330
      ISPLN=0                                                          00001340
46    TEM=TEM*DB                                                       00001350
      IF(TEM.GE.BMTEST) GO TO 47                                       00001360
      NS=NS+1                                                          00001370
      IF(NS.GT.200) CALL ERRMSG('SPLINED NS>200 IN FCODE',3,6,16)     00001380
      XS(NS)=TEM                                                       00001390
      YS(NS)=HZLOOP(TEM*TEM)                                           00001400
      GO TO 46                                                         00001410
47    CALL SPLIN1(NS,0.0,XS,YS,AS,BS,CS,0,DER,T,W2)                   00001420
      ISPLN=1                                                          00001430
```

```
49      TO=.5*TMIN/TCON                                             00001440
        TM=TMAX/TCON                                                00001450
        LATE=0                                                      00001460
        IF(IFIRST.EQ.1) IWARN=0                                     00001470
        NEW=1                                                       00001480
        TRANSL=1.E30                                                00001490
        DO 70 I=1,NN                                                00001500
        T(I)=X(I,1)/TCON                                            00001510
C--GET TRANSIENT IMPULSE RESPONSE VIA LAGGED CONVOLUTION IN TIME.   00001520
        TRANS=.63661977*RFLAGS(0,HZLOOP,EPS,TO,TM,T(I),NEW)         00001530
        NEW=0                                                       00001540
        IF(TRANS.GT.1.0) GO TO 71                                   00001550
C--IF CALC.TRANS TOO NOISY, THEN FORCE TRANS=TRANSL; THIS SHOULD NOT 00001560
C  OCCUR WITH THE USUAL TIME RANGE USED WITH MOST FIELD EQUIPMENT.  00001570
        IF(TRANS.LT.0.0.OR.TRANS.GT.TRANSL) THEN                    00001580
          TRANS=TRANSL                                              00001590
          IF(IWARN.EQ.0) THEN                                       00001600
            IWARN=1                                                 00001610
            CALL WARN('NOISE IN CALC. TRANS DETECTED.',0,6,16,*71)  00001620
          ENDIF                                                     00001630
        ENDIF                                                       00001640
71      TRANSL=TRANS                                                00001650
        VSAVE(I)=TRANS                                              00001660
70      CONTINUE                                                    00001670
C--IF IOPT=1, THEN CONVERT COMPUTED "TRANS" TO "APPRES"             00001680
        IF(IOPT.EQ.1) THEN                                          00001690
C**GET APP.RES. (SEE C**END OF "IF(IOPT.EQ.1) THEN" BELOW)          00001700
        DO 68 I=1,NN                                                00001710
          TIME=TCON*T(I)                                            00001720
          TV=T(I)*VSAVE(I)                                          00001730
C--MUELLERS ITER USING FCTCI(X1)=0 FOR SOLUTION X1 IN (0,20.)       00001740
        CALL DFIND(.1D-20,100,20.D0,FCTCI,XL,XR,IER)                00001750
        IF (IER.GT.0) THEN                                          00001760
          TAR(I,1)=1./SIG1                                          00001770
          X1=XR                                                     00001780
          TAR(I,2)=0.0                                              00001790
          GO TO 62                                                  00001800
        ENDIF                                                       00001810
        CALL DRTMI(X1,FX1,FCTCI,XL,XR,.1D-5,1000,IER)               00001820
        IF(IER.GT.0) THEN                                           00001830
          TAR(I,1)=1./SIG1                                          00001840
          X1=XL                                                     00001850
          TAR(I,2)=0.0                                              00001860
          GO TO 62                                                  00001870
        ENDIF                                                       00001880
C--X1 FOUND, GET APPRES TAR(I,1)                                    00001890
        TAR(I,1)=(XMU0*AA)/(4.*TIME*X1**2)                          00001900
        IF(TIME.GT.0.04) LATE=1                                     00001910
C--LOOK FOR 2ND X1 AND TAR(I,2), ETC.                               00001920
62      CALL DFIND(0.0D0,25,X1-.01*X1,FCTCI,XL,XR,IER)              00001930
        IF(IER.GT.0) THEN                                           00001940
          CALL DFIND(X1+.01*X1,25,1.0D5,FCTCI,XL,XR,IER)            00001950
          IF(IER.GT.0) THEN                                         00001960
            TAR(I,2)=0.0                                            00001970
            GO TO 68                                                00001980
```

```
         ENDIF                                                    00001990
        ENDIF                                                     00002000
        CALL DRTMI(X1,FX1,FCTCI,XL,XR,.1D-4,1000,IER)             00002010
        IF(IER.GT.0) THEN                                         00002020
          TAR(I,2)=0.0                                            00002030
          GO TO 68                                                00002040
        ENDIF                                                     00002050
        TAR(I,2)=(XMU0*AA)/(4.*TIME*X1**2)                        00002060
68      CONTINUE                                                  00002070
C--GET FINAL APPARENT RESISTIVITY APPRES(I) FROM TAR(I,2)         00002080
        TEM=1./SIG1                                               00002090
        IF(ABS(TAR(1,1)-TEM).LT.ABS(TAR(1,2)-TEM)) THEN           00002100
          APPRES(1)=TAR(1,1)                                      00002110
        ELSE                                                      00002120
          APPRES(1)=TAR(1,2)                                      00002130
        ENDIF                                                     00002140
        IF(APPRES(1).EQ.0.0) APPRES(1)=TEM                        00002150
        DO I=2,NN                                                 00002160
          J=I-1                                                   00002170
          IF(ABS(TAR(I,1)-APPRES(J)).LT.ABS(TAR(I,2)-APPRES(J))) THEN  00002180
            APPRES(I)=TAR(I,1)                                    00002190
          ELSE                                                    00002200
            APPRES(I)=TAR(I,2)                                    00002210
          ENDIF                                                   00002220
          IF(APPRES(I).EQ.0.0) APPRES(I)=APPRES(J)               00002230
        ENDDO                                                     00002240
        ENDIF                                                     00002250
C**END OF "IF(IOPT.EQ.1) THEN" ABOVE FOR APP.RES.                 00002260
        IF(ISTEP.EQ.1) THEN                                       00002270
C--GET STEP RESPONSE AS INTEGRAL OVER TIME OF IMPULSE RESPONSE.   00002280
        CALL INTEG1(NN,T,VSAVE,3.0)                               00002290
        ENDIF                                                     00002300
        IF(IDER.EQ.0) GO TO 600                                   00002310
        IFIRST=0                                                  00002320
        DO 80 J=1,M21                                             00002330
80      BSAVE(J)=B(J)                                             00002340
C--GET PRE-SPLINED TRANSIENT (EITHER ISTEP=0 OR 1) -OR- APPRES    00002350
600     IF(IOPT.EQ.0) THEN                                        00002360
            F=B(M2)*VSAVE(IN)/SIG1                                00002370
        ELSE                                                      00002380
          F=B(M2)*APPRES(IN)                                      00002390
        ENDIF                                                     00002400
        RETURN                                                    00002410
800     IF(IDER.EQ.0) GO TO 600                                   00002420
C--IDER=1 EST.DER.OPTION                                          00002430
8001    IF(IFIRST.EQ.1) GO TO 35                                  00002440
        DO 802 J=1,M21                                            00002450
          IF(B(J).NE.BSAVE(J)) GO TO 35                           00002460
802     CONTINUE                                                  00002470
        GO TO 600                                                 00002480
        END                                                       00002490
        REAL*8 FUNCTION FCTCI(X)                                  00002500
C--FUNCTION FCTCI(X) FOR ZERO OF FCTCI(X1)=0 VIA CALL DRTMI       00002510
        IMPLICIT REAL*8 (A-H,O-Z)                                 00002520
        PARAMETER (SQPI=1.772453850905516D0,CON1=2.D0/SQPI)       00002530
```

```
        COMMON/PASSER/TV,LATE                                           00002540
        X2=X*X                                                          00002550
        IF(LATE.EQ.0.OR.X.GE.0.1) THEN                                  00002560
        E=DEXP(-X2)                                                     00002570
        FCTCI=3.D0*DERF(X)-2.D0*X2*TV-CON1*X*(3.D0+2.D0*X2)*E            00002580
        ELSE                                                            00002590
          FCTCI=(((-2.D0*X2/33.D0+2.D0/9.D0)*X2-                        00002600
       1 4.D0/7.D0)*X2+0.8D0)*X2*X-SQPI*TV                              00002610
        ENDIF                                                           00002620
        RETURN                                                          00002630
        END                                                             00002640
        SUBROUTINE MARQ_TRANS_HZLOOP_SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT) 00002650
C-- INITIALIZATION ROUTINE (CALLED ONCE)                               00002660
C                                                                       00002670
C   SUBZ IS CALLED BY NLSOL AFTER THE DATA Y(I),X(I,5) ARE READ.        00002680
C   SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL SINIT                 00002690
C   PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...           00002700
C                                                                       00002710
C--PARAMETERS--                                                         00002720
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                         00002730
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY             00002740
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES              00002750
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                   00002760
C       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM IMSLMQ.                00002770
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                     00002780
C               0 IF ONLY UNIT 6 PRINT FILE USED.                       00002790
C                                                                       00002800
        CHARACTER*9 OPT(0:1)                                            00002810
        COMPLEX K2(10),KS1,C4,ZA,ZAC4                                   00002820
        CHARACTER*80 TITLE                                              00002830
        REAL Y(1),X(500,5),B(1),PRNT(1),SIG(10),H(10)                   00002840
        COMMON/PASS/ZAC4,ANORM,CURI,DC,SIG,BO,BM,SIG1,EPS,ISTEP         00002850
        COMMON/FPASS/AA,TMIN,TMAX,TO,TM,DB,BMTEST,                      00002860
       & M1,M21,M2,JSPLN,NN,IFIRST,IOPT                                 00002870
        COMMON/SPLN/FILL(1000),NS,ISPLN                                 00002880
        COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                              00002890
C**     NAMELIST/INIT/MM,A,Z,EPS,BO,BM,NB                               00002900
        COMMON/NAME_LIST/FILLS(65),MM,FILLS2(4),EPS_,                   00002910
       1 FILLER(3031),IOPT_,ISTEP_,NB,BO_,PARM(4),BM_,A_,Z_             00002920
        DATA ISUBZ/0/,OPT/'TRANSIENT','APPRES'/                        00002930
        IF(ISUBZ.NE.0) GO TO 10                                         00002940
C--PRESET                                                               00002950
        ISUBZ=1                                                         00002960
        MM=1                                                            00002970
        ISTEP_=0                                                        00002980
        A_=0.0                                                          00002990
        Z_=0.0                                                          00003000
        BO_=.01                                                         00003010
        BM_=100.                                                        00003020
        NB=8                                                            00003030
        EPS_=.1E-9                                                      00003040
        IOPT_=0                                                         00003050
C**10   READ(5,INIT)                                                    00003060
10      CALL NAMELIST(5,'SINIT',*11)                                    00003070
        IOPT=IOPT_                                                      00003080
```

```
        M=MM                                                            00003090
        EPS=EPS_                                                        00003100
        ISTEP=ISTEP_                                                    00003110
        B0=B0_                                                          00003120
        BM=BM_                                                          00003130
        A=A_                                                            00003140
        Z=Z_                                                            00003150
11      CALL NONBLANK(TITLE,NONBLK)                                     00003160
        WRITE(6,20) TITLE                                               00003170
20      FORMAT('1(NLSTCI):',5X,A<NONBLK>/)                             00003180
        IF(IOUT.EQ.1) WRITE(16,20) TITLE                               00003190
        WRITE(6,30) MM,A,EPS,B0,BM,NB,Z,ISTEP,IOPT                      00003200
        IF(IOUT.EQ.1) WRITE(16,30) MM,A,EPS,B0,BM,NB,Z,ISTEP,IOPT       00003210
30      FORMAT(' MM=',I3,12X,' A=',E13.6,4X,'EPS=',E13.6/              00003220
      &  ' B0=',E13.6,2X,'BM=',E13.6,4X,'NB=',I3/                       00003230
      &  ' Z=',E13.6,3X,'ISTEP=',I3/' IOPT=',I3)                        00003240
C--TEST SINIT PARMS                                                     00003250
        IF(MM.LT.1.OR.MM.GT.10.OR.A.LE.0.0.OR.NB.LT.0.OR.               00003260
      & ISTEP.LT.0.OR.ISTEP.GT.1.OR.IOPT.LT.0.OR.IOPT.GT.1.OR.          00003270
      & BM.LE.B0.OR.B0.LE.0.0.OR.Z.LT.0.0)                              00003280
      & CALL ERRMSG('SOME SINIT PARMS OUT OF RANGE.',6,6,16)            00003290
        IF(Z.GT.0.0.AND.ISTEP.EQ.1)                                    00003300
      1 CALL ERRMSG('Z>0 AND ISTEP=1 NOT ALLOWED.',1,6,16)             00003310
        IF(IOPT.EQ.1.AND.ISTEP.EQ.1)                                   00003320
      1 CALL ERRMSG('IOPT=1 AND ISTEP=1 NOT ALLOWED.',0,6,16)          00003330
C--TEST X(I, ) DATA BEFORE PROCEEDING                                   00003340
        DO 40 I=2,N                                                     00003350
        IF(X(I,1).LE.X(I-1,1).OR.X(I,1).LE.0.0)                         00003360
      & CALL ERRMSG('SOME X(I,1)<=0.0 OR NOT INCREASING.',7,6,16)       00003370
40      CONTINUE                                                        00003380
C--PRESET SOME GLOBAL CONSTANTS                                         00003390
        IFIRST=1                                                        00003400
        NN=N                                                            00003410
        AA=A*A                                                          00003420
        ZA=CMPLX(A,0.0)                                                 00003430
        CURI=.3183098861/AA                                            00003440
        C4=CMPLX(A/(2.0*SQRT(AA+Z*Z)**3),0.0)                          00003450
        ZAC4=ZA*C4                                                      00003460
        DC=A*CURI*REAL(C4)                                             00003470
        TMIN=X(1,1)                                                     00003480
        TMAX=X(N,1)                                                     00003490
        ISPLN=0                                                         00003500
        JSPLN=0                                                         00003510
        IF(NB.GT.0.AND.NB.LT.12) JSPLN=1                               00003520
        IF(JSPLN.EQ.1) THEN                                             00003530
         DB=EXP(2.30258509/FLOAT(NB))                                   00003540
         BMTEST=0.5*(BM+BM*DB)                                          00003550
        ENDIF                                                           00003560
        WRITE(6,50)                                                     00003570
        IF(IOUT.EQ.1) WRITE(16,50)                                      00003580
50      FORMAT(/////' PARAMETER ORDER--'/)                             00003590
        M1=MM-1                                                         00003600
        M21=2*MM-1                                                      00003610
        M2=M21+1                                                        00003620
        WRITE(6,110) (I,I,I=1,MM)                                       00003630
```

```
       IF(IOUT.EQ.1) WRITE(16,110) (I,I,I=1,MM)                     00003640
110    FORMAT(5X,I3,6X,6HSIGMA(,I3,1H))                            00003650
       IF(MM.EQ.1) GO TO 132                                       00003660
       DO 120 I=1,M1                                               00003670
       J=MM+I                                                      00003680
       IF(IOUT.EQ.1) WRITE(16,130) J,I                            00003690
120    WRITE(6,130) J,I                                            00003700
130    FORMAT(5X,I3,6X,6HTHICK(,I3,1H))                           00003710
132    WRITE(6,131) M2,M2,OPT(IOPT)                               00003720
131    FORMAT(5X,I3,10X,'B(',I3,') SHIFT PARAMETER IN B(2*MM)*',A) 00003730
       IF(IOUT.EQ.1) WRITE(16,131) M2,M2,OPT(IOPT)                00003740
       NPRNT=2                                                     00003750
       RETURN                                                      00003760
       END                                                        00003770
       SUBROUTINE NAMELIST(IUNIT,NAME,*)                           00003780
C                                                                  00003790
C {NAMELIST INPUT ON VAX-11/780} VIA "CALL NAMELIST" (VERSION: 12/10/80)00003800
C                                                                  00003810
C--A SIMULATED 'NAMELIST/NAME/' PROCESSOR FOR VAX-11 FORTRAN-77 TO  00003820
C  IMPLEMENT "CALL NAMELIST(IUNIT,'SNAME',*EOF)" ON VAX, WHICH     00003830
C  IS SIMILAR TO "READ(IUNIT,NAME,END=EOF)" ON MOST LARGE SYSTEMS. 00003840
C                                                                  00003850
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.      00003860
C                                                                  00003870
C--THIS IS A SUBSET OF THE ACTUAL NAMELIST/NAME/ AVAILABLE ON      00003880
C  MOST LARGE MAIN-FRAME SYSTEMS. CURRENT OPTIONS ARE:             00003890
C                                                                  00003900
C  (1) ALL VARNAM'S ARE RESTRICTED TO 1 TO 6 CHAR'S (ALP,NUM, AND '_') 00003910
C      BUT MUST BEGIN WITH AN ALP CHAR (E.G., A3_, BVAR, C_2, ETC.)   00003920
C  (2) ONLY VARIABLE TYPES REAL*4 *8 (NAMTYP=1) AND INTEGER*2 *4   00003930
C      (NAMTYP=0). SEE C==== EXAMPLE STATEMENTS FOR NAMTYP BELOW ====. 00003940
C      {NOTE: COMPLEX,LOGICAL, OR CHARACTER VARIABLE TYPES ARE "NOT"  00003950
C      CODED IN THIS VERSION.}                                     00003960
C  (3) MAX. 60 VARNAM'S ALLOWED IN NAMELIST (FOR ALL 'SNAMES' USED). 00003970
C  (4) MAX. NUMBER FIELD (FLOAT OR FIXED) IS 20 CHAR WIDE, WHERE   00003980
C      BLANK CHAR'S ARE IGNORED, AND TYPE CONVERSION IS AUTOMATIC. 00003990
C      FLOAT NUMBERS WITH OPTIONAL E+XX OR D=XX AND WITH OR WITHOUT '.' 00004000
C      IN THE MANTISSA IS ALLOWED (E.G., 123E-3, .123D+02, -3.14, ETC.).00004010
C  (5) PARTIAL ARRAY'S ALLOWED; E.G., A(10)=25.1,                  00004020
C      AND B=1,3.2,...                                             00004030
C  (6) REPEAT FACTORS ALLOWED; E.G., C=2*1,3,...                   00004040
C  (7) ONLY 1-DIM ARRAYS ALLOWED WITH MAX SIZE 99999.              00004050
C  (8) THE NAMELIST 'SNAME' MUST BE 2 TO 7 CHAR'S, AND MUST BEGIN WITH 00004060
C      A "S" CHAR (E.G., 'SP', 'SPARMS', ETC.); ALSO, THE FIRST CHAR IN00004070
C      IFILE MAY BEGIN IN COL. 1 BUT LESS THAN COL. 72 (BUFFER IS 80). 00004080
C      LINES IN IFILE MAY BE CONTINUED TO COL. 1 ON NEXT LINE, AND 00004090
C      TERMINATE THE NAMELIST BY "S[END]"--THE "END" IS OPTIONAL. E.G.,00004100
C                                                                  00004110
C      SPARMS A=1,B=2.3,7*1,C(3)=-.123E-10,                       00004120
C      D=1800, E=5*20SEND                                          00004130
C      SNEXNAM F=123, G=-10,C(2)=15.02 S                          00004140
C      ...END-OF-IFILE...                                          00004150
C  (9) ABOUT 98% OF ALL THE POSSIBLE ERRORS ARE DETECTED AND AN    00004160
C      ERROR MESSAGE IS PRINTED ON UNIT 06, FOLLOWED BY CALL EXIT. 00004170
C      {NOTE: WATCH OUT FOR THE REMAINING 2% UNDETECTED ERRORS!}   00004180
```

```
C                                                                   00004190
C--SUBROUTINES CALLED:                                              00004200
C                                                                   00004210
C   DECODEIX, DECODEX, AND NONBLANK.                                00004220
C                                                                   00004230
C--USAGE:                                                           00004240
C                                                                   00004250
C  1. MODIFY FILE 'INCLNAMES.FOR' AS REQUIRED (USE ANY EDITOR).     00004260
C     (SEE C==== EXAMPLE STATEMENTS BELOW ====.)                    00004270
C  2. RECOMPILE SUBROUTINE 'NAMELIST' WITH THE DESIRED INCLNAMES.FOR. 00004280
C  3. IN USERS CALLING PROGRAM, USE:                               00004290
C     CALL NAMELIST(IUNIT,'SNAME',*N)   --ON VAX, WHERE N=E.O.F RETURN 00004300
C     STATEMENT LABEL.  THIS SIMULATES ON VAX:                     00004310
C      'READ(IUNIT,NAME,END=N)' ON SYSTEMS WITH NAMELIST/NAME/...  00004320
C                                                                   00004330
C*****************************************************************00004340
C                                                                   00004350
      CHARACTER*(*) NAME                                           00004360
      CHARACTER*1 C(47),BUFI                                        00004370
      CHARACTER*6 VARNAM                                            00004380
      CHARACTER*20 NUMFLD                                           00004390
      CHARACTER*80 BUF                                              00004400
C                                                                   00004410
C===========================================================00004420
C====== THE USER MUST CHANGE THE FOLLOWING STATEMENTS FOR THE SPECIFIC 00004430
C====== NAMELIST VARIABLES DESIRED (E.G., USE TECO OR EDT, ETC.)======00004440
C====== DIMENSION NO_NAM VARIABLES TO AGREE WITH CHANGED DATA STATEMENTS00004450
C==                                                                 00004460
C==ON VAX USE THE FOLLOWING INCLUDE STATEMENT (OPTIONALLY, USE /LIST): 00004470
C==                                                                 00004480
C>>   INCLUDE 'INCLNAMES.FOR/NOLIST'                                00004490
C                                                                   00004500
C================== INCLNAM13.FT =======================00004510
C================ FOR USE IN CALL NAMELIST ================00004520
C NORMALLY, ONE SHOULD COPY 'INCLNAM13.FT' TO 'INCLNAMES.FT'; THEN  00004530
C   EDIT 'INCLNAMES.FT' AS DESIRED FOR USERS CALL NAMELIST. NOTE THAT 00004540
C   ONE MUST RECOMPILE 'NAMELIST.FT' WITH USERS CALLING PROGRAM,    00004550
C   WHERE 'NAMELIST.FT' CONTAINS THE FOLLOWING STATEMENT:           00004560
C                                                                   00004570
C     INCLUDE 'INCLNAMES.FT/LIST'                                   00004580
C===========================================================00004590
C                                                                   00004600
C*****************************************************************00004610
C   THIS IS 'SPARMS AND SINIT' INPUT FOR PROGRAMS "NLSTCI" AND "NLSTCO" 00004620
C*****************************************************************00004630
C                                                                   00004640
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00004650
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF     00004660
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                    00004670
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                          00004680
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS. 00004690
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:        00004700
      PARAMETER (K1DIM=KDIM-1,                                      00004710
     1 IVDIM=KDIM+60,NKVDIM=96+2*NDIM+(KDIM*(7*KDIM+41))/2)        00004720
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00004730
```

```
C                                                                  00004740
      COMMON/NAME_LIST/V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,              00004750
     * V11,V12,V13,V14,V15,V16,V17,V18,V19,V20,                     00004760
     * V21,V22,V23,V24,V25,V26,V27,V28,V29,V30,                     00004770
     * V31,V32,V33,V34,V35,V36,V37,V38,V39,                         00004780
     * V40,V41,V42,V43,V44,V45,V46,V47,V48,V49,V50,V51              00004790
      INTEGER V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,                   00004800
     * V17, V21,V22,V23,V24,V25, V27,V28,V29, V35,V36,V37,V38,V39,  00004810
     * V40,V44,V45,V46                                              00004820
      DIMENSION V1(1),V2(1),V3(1),V4(1),                            00004830
     * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),                        00004840
     * V11(1),V12(1),V13(1),V14(1),V15(1),                          00004850
     * V16(1),V17(1),V18(1),V19(1),V20(1),                          00004860
     * V21(1),V22(1),V23(1),V24(1),V25(1),                          00004870
     * V26(KDIM),V27(K1DIM),V28(1),V29(1),V30(1),                   00004880
     * V31(1),V32(1),V33(1),V34(1),V35(1),                          00004890
     * V36(1),V37(1),V38(1),V39(1),V40(IVDIM),                      00004900
     * V41(NKVDIM),V42(KDIM),V43(KDIM),V44(1),V45(1),               00004910
     * V46(1),V47(1),V48(4),V49(1),V50(2),                          00004920
     * V51(1),V52(1),V53(1),V54(1),V55(1),                          00004930
     * V56(1),V57(1),V58(1),V59(1),V60(1)                           00004940
      DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)                    00004950
      CHARACTER*6 NAM(60)                                           00004960
      DATA NAM/'N','K','IP','M','IALT','ISTOP','IWT','IDER',        00004970
     * 'IPRT','NITER','INON','FF','T','E','TAU','XL','MODLAM',      00004980
     * 'GAMCR','DEL','ZETA','IOUT','SP','SCALEP','SY','SCALEY',     00004990
     * 'B','IB','IOB','MM','X0','Y0','L','EP','EPS','NEPS',         00005000
     * 'METHOD','NFIN','IER','MEV','IV','V','BL','BH',              00005010
     * 'IOPT','ISTEP','NB','B0','PARM','BM','A','Z',9*' '/          00005020
      DATA NAMDIM/25*1,KDIM,K1DIM,12*1,IVDIM,NKVDIM,2*KDIM,4*1,     00005030
     1 4,3*1,9*0/                                                   00005040
      DATA NAMLEN/2*1,2,1,4,5,3,2*4,5,4,2,2*1,3,2,6,5,3,2*4,        00005050
     * 2,6,2,6,1,2,3,3*2,1,2,3,4,6,4,2*3,2,1,2*2,                   00005060
     * 4,5,2*2,4,2,2*1,9*0/                                         00005070
      DATA NAMTYP/11*0,5*1,0,3*1,5*0,1,3*0,5*1,5*0,0,3*1,3*0,5*1,9*0/  00005080
      DATA NO_NAM/51/                                               00005090
C====== END OF INCLUDE STATEMENTS ==================================00005100
C                                                                  00005110
C==                                                                00005120
C== FOR EXAMPLE, FILE 'INCLNAMES.FOR' MAY CONTAIN (WITHOUT "C=="): 00005130
C==                                                                00005140
C==       COMMON/NAME_LIST/V1,V2,V3,V4                             00005150
C==       REAL*8 V1                                                00005160
C==       INTEGER V3                                               00005170
C==       DIMENSION V1(1),V2(2),V3(3),V4(4),                       00005180
C==      * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),                   00005190
C==      * V11(1),V12(1),V13(1),V14(1),V15(1),                     00005200
C==      * V16(1),V17(1),V18(1),V19(1),V20(1),                     00005210
C==      * V21(1),V22(1),V23(1),V24(1),V25(1),                     00005220
C==      * V26(1),V27(1),V28(1),V29(1),V30(1),                     00005230
C==      * V31(1),V32(1),V33(1),V34(1),V35(1),                     00005240
C==      * V36(1),V37(1),V38(1),V39(1),V40(1),                     00005250
C==      * V41(1),V42(1),V43(1),V44(1),V45(1),                     00005260
C==      * V46(1),V47(1),V48(1),V49(1),V50(1),                     00005270
C==      * V51(1),V52(1),V53(1),V54(1),V55(1),                     00005280
```

```
C==        * V56(1),V57(1),V58(1),V59(1),V60(1)                       00005290
C==        DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)                 00005300
C==        CHARACTER*6 NAM(60)                                        00005310
C==        DATA NAM/'A','BB','ICC','DDD_4',56*' '/                    00005320
C==        DATA NAMDIM/1,2,3,4,56*0/                                  00005330
C==        DATA NAMLEN/1,2,3,5,56*0/                                  00005340
C==        DATA NAMTYP/2*1,0,1,56*0/                                  00005350
C==        DATA NO_NAM/4/                                             00005360
C====== END OF EXAMPLE INCLUDE STATEMENTS ======================00005370
C                                                                     00005380
C****************************************************************00005390
C NOTE: THE ABOVE EXAMPLE SIMULATES                                   00005400
C         'NAMELIST/NAME/A,BB,ICC,DDD_4'                              00005410
C         'READ(IUNIT,NAME,END=EOF)'                                  00005420
C         'READ(IUNIT,ANYNAM,END=EOF)'                                00005430
C         IN THE CALLING PROGRAM USING:                              00005440
C         ...                                                         00005450
C         REAL*8 A                                                    00005460
C         ...                                                         00005470
C         COMMON/NAME_LIST/A,BB(2),ICC(3),DDD_4(4)                    00005480
C         ...                                                         00005490
C         CALL NAMELIST(IUNIT,'$NAME',*EOF)                           00005500
C         ...                                                         00005510
C         CALL NAMELIST(IUNIT,'$ANYNAM',*EOF)                         00005520
C         ...                                                         00005530
C****************************************************************00005540
C                                                                     00005550
      DATA C/'A','B','C','D','E','F','G','H','I','J','K','L','M','N',  00005560
     * 'O','P','Q','R','S','T','U','V','W','X','Y','Z','_',            00005570
     * '1','2','3','4','5','6','7','8','9','0',                        00005580
     * ' ','$','=',',','(','*',')','.','+','-'/                        00005590
      J=LEN(NAME)                                                      00005600
      IF(J.LT.2.OR.J.GT.7) THEN                                        00005610
        CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//             00005620
     1 NAME//'    (LENGTH<2 OR >7 CHAR''S)',1,6,0)                     00005630
      ENDIF                                                            00005640
      IF(NAME(1:1).NE.'$')                                            00005650
     1 CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//              00005660
     2 NAME//'    (1ST CHAR MUST BE "$" CHAR)',1,6,0)                  00005670
C--INITIALIZE                                                         00005680
      INAME=0                                                          00005690
10    READ(IUNIT,11,END=99991,ERR=99992) BUF                          00005700
11    FORMAT(A80)                                                     00005710
      IF(INAME.EQ.1) GO TO 20                                          00005720
C--LOOK FOR '$NAME'                                                   00005730
      I=INDEX(BUF,NAME)                                                00005740
      IF(I.EQ.0) GO TO 10                                              00005750
      INAME=1                                                          00005760
      ICOL=I+J                                                         00005770
      JNAM=0                                                           00005780
      ILEN=0                                                           00005790
      VARNAM=' '                                                       00005800
      NUMLEN=0                                                         00005810
      IELE=1                                                           00005820
      GO TO 30                                                         00005830
```

```
20      ICOL=1                                                      00005840
30      CALL NONBLANK(BUF,LENBUF)                                   00005850
C==BEGIN PARSER LOOP (THE BIG 20000 LOOP)                           00005860
        IEND=0                                                      00005870
        DO 20000 I=ICOL,LENBUF                                      00005880
            BUFI=BUF(I:I)                                           00005890
        DO 40 IC=1,27                                               00005900
            IF(BUFI.EQ.C(IC)) GO TO 100                             00005910
40      CONTINUE                                                    00005920
        DO 50 IC=28,37                                              00005930
        IF(BUFI.EQ.C(IC)) GO TO 200                                 00005940
50      CONTINUE                                                    00005950
        DO 60 IC=38,47                                              00005960
        IC_=IC-37                                                   00005970
        IF(BUFI.EQ.C(IC)) GO TO 70                                  00005980
60      CONTINUE                                                    00005990
61      WRITE(6,66) I,BUF                                           00006000
66      FORMAT(/' (NAMELIST): ERROR IN FOLLOWING RECORD AT COL(',I2,'):'/  00006010
1 1X,A80/<I>X,'^')                                                  00006020
        CALL ERRMSG('ILLEGAL CHAR=''//BUFI//'" FOUND',0,6,0)        00006030
67      WRITE(6,66) I,BUF                                           00006040
        CALL ERRMSG('NUMLEN<1 IN DECODEIX    ',0,6,0)               00006050
68      WRITE(6,66) I,BUF                                           00006060
        CALL ERRMSG('NUMLEN<1 IN DECODEX',0,6,0)                    00006070
70      GO TO (20000,72,73,74,75,76,77,78,79,79),IC_               00006080
C--'S' CHAR                                                         00006090
72      IEND=1                                                      00006100
        IF(NUMLEN.GT.0) GO TO 798                                   00006110
        IF(JNAM.EQ.0) GO TO 99990                                   00006120
        WRITE(6,66) I,BUF                                           00006130
        CALL ERRMSG('MISPLACED "s" CHAR',0,6,0)                     00006140
C--'=' CHAR                                                         00006150
73      IEQ=1                                                       00006160
C--CHECK FOR VALID VARNAM, LENGTH ILEN, ETC.                        00006170
        IF(ILEN.LT.1) GO TO 733                                     00006180
        DO 732 J=1,NO_NAM                                           00006190
        JNAM=J                                                      00006200
        JLEN=NAMLEN(J)                                              00006210
        IF(JLEN.NE.ILEN) GO TO 732                                  00006220
        DO 731 K=1,JLEN                                             00006230
        IF(VARNAM(K:K).NE.NAM(JNAM)(K:K)) GO TO 732                 00006240
731     CONTINUE                                                    00006250
C--VARNAM VERIFIED OK TO PROCEED TO NUMFLD(S)                       00006260
C                                                                   00006270
        IDIM=NAMDIM(JNAM)                                           00006280
        NUMLEN=0                                                    00006290
        NDEC=0                                                      00006300
        NREP=1                                                      00006310
        NEXP=0                                                      00006320
        GO TO 20000                                                 00006330
732     CONTINUE                                                    00006340
        WRITE(6,66) I,BUF                                           00006350
        CALL ERRMSG('ILLEGAL VARNAM='//VARNAM//' FOUND',0,6,0)      00006360
733     WRITE(6,66) I,BUF                                           00006370
        CALL ERRMSG('MISPLACED "=" CHAR   ',0,6,0)                  00006380
```

```
C--',' CHAR                                                           00006390
74     IF(NUMLEN.GT.0) GO TO 799                                      00006400
       WRITE(6,66) I,BUF                                              00006410
       CALL ERRMSG('MISPLACED "," CHAR',0,6,0)                        00006420
C--'(' CHAR                                                           00006430
75     IELE=0                                                         00006440
       GO TO 20000                                                    00006450
C--'*' CHAR                                                           00006460
76     IF(JNAM.EQ.0.OR.NUMLEN.LT.1.OR.NUMLEN.GT.5) GO TO 767          00006470
760    CALL DECODEIX(NUMFLD,NUMLEN,NREP,*67)                          00006480
       NUMLEN=0                                                       00006490
       IF(NREP.GT.0.AND.NREP.LE.NAMDIM(JNAM)) GO TO 20000             00006500
       WRITE(6,66) I,BUF                                              00006510
       CALL ERRMSG('REPEAT FACTOR <1 OR >NAMDIM   ',0,6,0)            00006520
767    WRITE(6,66) I,BUF                                              00006530
       CALL ERRMSG('REPEAT WIDTH > 5 OR MISPLACED "*" CHAR',0,6,0)    00006540
C--')' CHAR                                                           00006550
77     IF(IELE.NE.0) GO TO 772                                        00006560
       CALL DECODEIX(NUMFLD,NUMLEN,IELE,*67)                          00006570
       IF(IELE.LT.1) GO TO 773                                        00006580
       NREP=1                                                         00006590
       GO TO 20000                                                    00006600
772    WRITE(6,66) I,BUF                                              00006610
       CALL ERRMSG('MISPLACED ")" CHAR',0,6,0)                        00006620
773    WRITE(6,66) I,BUF                                              00006630
       CALL ERRMSG('ARRAY IELE<1 OR >NAMDIM   ',0,6,0)                00006640
C--'.' CHAR                                                           00006650
78     IF(JNAM.EQ.0.OR.NEXP.GT.0.OR.NDEC.GT.0) GO TO 781              00006660
       NDEC=NUMLEN+1                                                  00006670
       IF(NAMTYP(JNAM).EQ.1) GO TO 200                                00006680
781    WRITE(6,66) I,BUF                                              00006690
       CALL ERRMSG('MISPLACED "." CHAR',0,6,0)                        00006700
C--'-' OR '+' CHAR                                                    00006710
79     IF(IELE.GT.0.OR.NEXP.GT.0) GO TO 210                           00006720
       WRITE(6,66) I,BUF                                              00006730
       CALL ERRMSG('MISPLACED "-" OR "+" CHAR',0,6,0)                 00006740
C--<ALP> CHAR                                                         00006750
100    IF(NUMLEN.GT.0) GO TO 209                                      00006760
       IF(ILEN.GT.0) GO TO 102                                        00006770
       IEQ=0                                                          00006780
       IELE=1                                                         00006790
102    ILEN=ILEN+1                                                    00006800
       IF(ILEN.GT.6) GO TO 101                                        00006810
       VARNAM(ILEN:ILEN)=BUFI                                         00006820
       GO TO 20000                                                    00006830
101    WRITE(6,66) I,BUF                                              00006840
       CALL ERRMSG('VARNAM>6 CHAR''S',0,6,0)                          00006850
C--<+-NUM> CHAR                                                       00006860
200    IF(IELE.EQ.0) GO TO 210                                        00006870
       IF(IEQ.EQ.0) GO TO 102                                         00006880
       GO TO 210                                                      00006890
209    IF(BUFI.EQ.'E'.OR.BUFI.EQ.'D') THEN                           00006900
         NEXP=NUMLEN+1                                                00006910
       ELSE                                                           00006920
         GO TO 61                                                     00006930
```

```
        ENDIF                                                   00006940
210     NUMLEN=NUMLEN+1                                         00006950
        IF(NUMLEN.GT.20) GO TO 211                              00006960
        NUMFLD(NUMLEN:NUMLEN)=BUFI                              00006970
        GO TO 20000                                            00006980
211     WRITE(6,66) I,BUF                                       00006990
        CALL ERRMSG('NUM FIELD>20 CHAR''S',0,6,0)              00007000
C--PROCESS NUMBER FIELD                                         00007010
799     IDIM=IDIM-1                                             00007020
        IF(IDIM.LT.0) GO TO 10004                               00007030
798     IF(NEXP.GT.0) GO TO 1000                                00007040
C--[NEXP=0]                                                     00007050
        IF(NDEC.GT.0) GO TO 899                                 00007060
C--[NEXP=0, NDEC=0]                                             00007070
        CALL DECODEIX(NUMFLD,NUMLEN,IX,*67)                     00007080
C--CONVERT IX AND STORE IN COMMON                               00007090
800     X=IX                                                   00007100
        IF(IELE.GT.NAMDIM(JNAM)) GO TO 773                      00007110
8000    GO TO (801,802,803,804,805,806,807,808,809,810,        00007120
     *  811,812,813,814,815,816,817,818,819,820,               00007130
     *  821,822,823,824,825,826,827,828,829,830,               00007140
     *  831,832,833,834,835,836,837,838,839,840,               00007150
     *  841,842,843,844,845,846,847,848,849,850,               00007160
     *  851,852,853,854,855,856,857,858,859,860),JNAM          00007170
801     V1(IELE)=X                                             00007180
        GO TO 10000                                            00007190
802     V2(IELE)=X                                             00007200
        GO TO 10000                                            00007210
803     V3(IELE)=X                                             00007220
        GO TO 10000                                            00007230
804     V4(IELE)=X                                             00007240
        GO TO 10000                                            00007250
805     V5(IELE)=X                                             00007260
        GO TO 10000                                            00007270
806     V6(IELE)=X                                             00007280
        GO TO 10000                                            00007290
807     V7(IELE)=X                                             00007300
        GO TO 10000                                            00007310
808     V8(IELE)=X                                             00007320
        GO TO 10000                                            00007330
809     V9(IELE)=X                                             00007340
        GO TO 10000                                            00007350
810     V10(IELE)=X                                            00007360
        GO TO 10000                                            00007370
811     V11(IELE)=X                                            00007380
        GO TO 10000                                            00007390
812     V12(IELE)=X                                            00007400
        GO TO 10000                                            00007410
813     V13(IELE)=X                                            00007420
        GO TO 10000                                            00007430
814     V14(IELE)=X                                            00007440
        GO TO 10000                                            00007450
815     V15(IELE)=X                                            00007460
        GO TO 10000                                            00007470
816     V16(IELE)=X                                            00007480
```

```
              GO TO 10000                                     00007490
     817      V17(IELE)=X                                     00007500
              GO TO 10000                                     00007510
     818      V18(IELE)=X                                     00007520
              GO TO 10000                                     00007530
     819      V19(IELE)=X                                     00007540
              GO TO 10000                                     00007550
     820      V20(IELE)=X                                     00007560
              GO TO 10000                                     00007570
     821      V21(IELE)=X                                     00007580
              GO TO 10000                                     00007590
     822      V22(IELE)=X                                     00007600
              GO TO 10000                                     00007610
     823      V23(IELE)=X                                     00007620
              GO TO 10000                                     00007630
     824      V24(IELE)=X                                     00007640
              GO TO 10000                                     00007650
     825      V25(IELE)=X                                     00007660
              GO TO 10000                                     00007670
     826      V26(IELE)=X                                     00007680
              GO TO 10000                                     00007690
     827      V27(IELE)=X                                     00007700
              GO TO 10000                                     00007710
     828      V28(IELE)=X                                     00007720
              GO TO 10000                                     00007730
     829      V29(IELE)=X                                     00007740
              GO TO 10000                                     00007750
     830      V30(IELE)=X                                     00007760
              GO TO 10000                                     00007770
     831      V31(IELE)=X                                     00007780
              GO TO 10000                                     00007790
     832      V32(IELE)=X                                     00007800
              GO TO 10000                                     00007810
     833      V33(IELE)=X                                     00007820
              GO TO 10000                                     00007830
     834      V34(IELE)=X                                     00007840
              GO TO 10000                                     00007850
     835      V35(IELE)=X                                     00007860
              GO TO 10000                                     00007870
     836      V36(IELE)=X                                     00007880
              GO TO 10000                                     00007890
     837      V37(IELE)=X                                     00007900
              GO TO 10000                                     00007910
     838      V38(IELE)=X                                     00007920
              GO TO 10000                                     00007930
     839      V39(IELE)=X                                     00007940
              GO TO 10000                                     00007950
     840      V40(IELE)=X                                     00007960
              GO TO 10000                                     00007970
     841      V41(IELE)=X                                     00007980
              GO TO 10000                                     00007990
     842      V42(IELE)=X                                     00008000
              GO TO 10000                                     00008010
     843      V43(IELE)=X                                     00008020
              GO TO 10000                                     00008030
```

```
844    V44(IELE)=X                                                  00008040
       GO TO 10000                                                  00008050
845    V45(IELE)=X                                                  00008060
       GO TO 10000                                                  00008070
846    V46(IELE)=X                                                  00008080
       GO TO 10000                                                  00008090
847    V47(IELE)=X                                                  00008100
       GO TO 10000                                                  00008110
848    V48(IELE)=X                                                  00008120
       GO TO 10000                                                  00008130
849    V49(IELE)=X                                                  00008140
       GO TO 10000                                                  00008150
850    V50(IELE)=X                                                  00008160
       GO TO 10000                                                  00008170
851    V51(IELE)=X                                                  00008180
       GO TO 10000                                                  00008190
852    V52(IELE)=X                                                  00008200
       GO TO 10000                                                  00008210
853    V53(IELE)=X                                                  00008220
       GO TO 10000                                                  00008230
854    V54(IELE)=X                                                  00008240
       GO TO 10000                                                  00008250
855    V55(IELE)=X                                                  00008260
       GO TO 10000                                                  00008270
856    V56(IELE)=X                                                  00008280
       GO TO 10000                                                  00008290
857    V57(IELE)=X                                                  00008300
       GO TO 10000                                                  00008310
858    V58(IELE)=X                                                  00008320
       GO TO 10000                                                  00008330
859    V59(IELE)=X                                                  00008340
       GO TO 10000                                                  00008350
860    V60(IELE)=X                                                  00008360
       GO TO 10000                                                  00008370
C--[NEXP=0, NDEC>0]                                                 00008380
899    CALL DECODEX(NUMFLD,NUMLEN,NDEC,X,*68)                       00008390
C--CONVERT X AND STORE IN COMMON                                    00008400
900    IF(IELE.GT.NAMDIM(JNAM)) GO TO 773                           00008410
       GO TO 8000                                                   00008420
C--[NEXP>0]                                                         00008430
1000   IF(NDEC.GT.0) GO TO 2000                                     00008440
C--[NEXP>0, NDEC=0]                                                 00008450
       CALL DECODEIX(NUMFLD,NEXP-1,IX,*67)                          00008460
       X=IX                                                         00008470
1002   J=1                                                          00008480
       DO 1001 K=NEXP+1,NUMLEN                                      00008490
       NUMFLD(J:J)=NUMFLD(K:K)                                      00008500
1001   J=J+1                                                        00008510
       CALL DECODEIX(NUMFLD,NUMLEN-NEXP,IE,*67)                     00008520
       X=X*10.**IE                                                  00008530
C** (LATER INSERT A CALL TO A OVERFLOW HANDLER, ETC.)               00008540
       GO TO 900                                                    00008550
C--[NEXP>0, NDEC>0]                                                 00008560
2000   CALL DECODEX(NUMFLD,NEXP-1,NDEC,X,*68)                       00008570
       GO TO 1002                                                   00008580
```

```
C--NEXT IELE?                                                     00008590
10000 IELE=IELE+1                                                 00008600
      IF(IELE.GT.NAMDIM(JNAM)) GO TO 10002                        00008610
      IF(NREP.GT.1) GO TO 10003                                   00008620
10001 IF(IEND.EQ.1) GO TO 99990                                   00008630
      NUMLEN=0                                                    00008640
      NDEC=0                                                      00008650
      NEXP=0                                                      00008660
      NREP=1                                                      00008670
      ILEN=0                                                      00008680
      VARNAM=' '                                                  00008690
      GO TO 20000                                                 00008700
10002 IELE=1                                                      00008710
      GO TO 10001                                                 00008720
10003 NREP=NREP-1                                                 00008730
      IDIM=IDIM-1                                                 00008740
      IF(IDIM.GE.0) GO TO 8000                                    00008750
10004 WRITE(6,66) I,BUF                                           00008760
      CALL ERRMSG('TOO MANY ELEMENTS FOR GIVEN NAMDIM.',0,6,0)    00008770
C==END OF DO 20000   CONTINUE PARSER -OR- READ IN NEXT BUF, ETC.  00008780
20000 CONTINUE                                                    00008790
                GO TO 10                                          00008800
C--'S' CHAR (DELIMITER S[END] FOR THIS $NAME --S)                 00008810
99990 RETURN                                                      00008820
C--E.O.F. ON FILE IUNIT ENCOUNTERED.                              00008830
99991 RETURN 1                                                    00008840
99992 CALL ERRMSG('CANNOT OPEN/READ CALL NAMELIST(IFILE,...)',1,6,0) 00008850
      END                                                        00008860
      SUBROUTINE DUMYPCODE()                                      00008870
C--DUMMY PCODE FOR USE IN 'MARQRT' OR 'NLSOL'                     00008880
      CALL ERRMSG('IDER=0 NOT AVAILABLE IN THIS VERSION.',4,6,16) 00008890
      END                                                        00008900
      SUBROUTINE SIGSUBEND(Y,X,B,K,N,TITLE,IOUT)                  00008910
C**GENERAL SUBEND TERMINATION ROUTINE WITH 'SIGMA' NAMES.         00008920
C  ALSO GIVES RESTART SPARMS ON UNIT=4 AS 'FOR005.TMP'            00008930
C                                                                 00008940
      CHARACTER*132 LINE                                          00008950
      CHARACTER*80 TITLE                                          00008960
      REAL Y(1),X(500,5),B(1)                                     00008970
      CALL NONBLANK(TITLE,NB)                                     00008980
      WRITE(6,10) TITLE                                           00008990
10    FORMAT(//' ********  E N D  ********',5X,A<NB>//            00009000
     1 ' PARAMETER NAME',6X,'FINAL SOLUTION',8X,                  00009010
     2 'RESISTIVITY   LAYER DEPTH'/)                              00009020
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                           00009030
      MM=(K+1)/2                                                  00009040
      DO 30 I=1,MM                                                00009050
      R=1.0/B(I)                                                  00009060
      WRITE(6,20) I,I,B(I),I,R                                    00009070
20    FORMAT(2X,I3,3X,'SIGMA(',I2,') =',E16.8,2X,I2,E16.8)        00009080
      IF(IOUT.EQ.1) WRITE(16,20) I,I,B(I),I,R                     00009090
30    CONTINUE                                                    00009100
      K1=0                                                        00009110
      IF(K.EQ.1) GO TO 60                                         00009120
      IF(K.EQ.2) GO TO 52                                         00009130
```

```
        M2=MM+1                                                      00009140
        K1=K                                                         00009150
        IF(MOD(K,2).EQ.0) K1=K-1                                     00009160
        D=0.0                                                        00009170
        DO 50 I=M2,K1                                                00009180
        D=D+B(I)                                                     00009190
        L=I-MM                                                       00009200
        WRITE(6,40) I,L,B(I),L,D                                     00009210
40      FORMAT(2X,I3,3X,'THICK(',I2,') =',E16.8,22X,I2,E16.8)        00009220
        IF(IOUT.EQ.1) WRITE(16,40) I,L,B(I),L,D                      00009230
50      CONTINUE                                                     00009240
        IF(K1.EQ.K) GO TO 60                                         00009250
52      WRITE(6,54) K,B(K)                                           00009260
54      FORMAT(2X,I3,3X,'SHIFT',5X,'=',E16.8)                        00009270
        IF(IOUT.EQ.1) WRITE(16,54) K,B(K)                            00009280
C** GENERATE RESTART SPARMS ON FOR005.TMP                            00009290
60      REWIND 5                                                     00009300
        OPEN(UNIT=4,FILE='FOR005.TMP',STATUS='NEW',                  00009310
     1 CARRIAGECONTROL='LIST')                                       00009320
        READ(5,65,END=999) LINE                                      00009330
65      FORMAT(A)                                                    00009340
        CALL NONBLANK(LINE,NB)                                       00009350
        WRITE(4,66) LINE                                             00009360
66      FORMAT(A<NB>)                                                00009370
        IDOL=0                                                       00009380
70      READ(5,65,END=999) LINE                                      00009390
        I=INDEX(LINE,'$')                                            00009400
        IF(I.NE.0) THEN                                              00009410
          IF(IDOL.EQ.0) THEN                                         00009420
            IDOL=1                                                   00009430
            J=INDEX(LINE(I+1:),'$')                                  00009440
            IF(J.NE.0) THEN                                          00009450
              IDOL=2                                                 00009460
              LINE(J:J)=','                                          00009470
            ENDIF                                                    00009480
          ELSE                                                       00009490
            IDOL=2                                                   00009500
            LINE(I:I)=','                                            00009510
          ENDIF                                                      00009520
        ENDIF                                                        00009530
        CALL NONBLANK(LINE,NB)                                       00009540
        WRITE(4,66) LINE                                             00009550
        IF(IDOL.LT.2) GO TO 70                                       00009560
        LINE(1:)='B='                                                00009570
        DO 80 I=1,K                                                  00009580
        ENCODE(16,90,LINE(3:18)) B(I)                                00009590
90      FORMAT(G16.8)                                                00009600
        IF(I.LT.K) THEN                                              00009610
          LINE(19:19)=','                                           00009620
        ELSE                                                         00009630
          LINE(19:19)='$'                                            00009640
        ENDIF                                                        00009650
        CALL NONBLANK(LINE,NB)                                       00009660
        WRITE(4,66) LINE                                             00009670
        LINE(1:2)='  '                                               00009680
```

```
80      CONTINUE                                                  00009690
100     READ(5,65,END=999) LINE                                   00009700
        CALL NONBLANK(LINE,NB)                                    00009710
        WRITE(4,66) LINE                                          00009720
        GO TO 100                                                 00009730
999     RETURN                                                    00009740
        END                                                       00009750
        SUBROUTINE CPUTIME(I1,I2)                                 00009760
C                                                                 00009770
C  CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON 00009780
C  FORTRAN UNITS I1 (IF NOT 0) AND I2 (IF NOT 0).                 00009790
C                                                                 00009800
C  WILL EJECT FIRST IF I1>0 (OR I2>0).                            00009810
C  DOUBLE SPACE FIRST IF I1<0 (OR I2<0).                          00009820
C                                                                 00009830
C  E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS:00009840
C                                                                 00009850
C      CALL SETTIME    ! DON'T FORGET TO DO THIS!                 00009860
C      >>>>> THE CODE TO TIME IS HERE <<<<<  ! USUALLY A COMPLETE PROGRAM00009870
C      CALL CPUTIME(-6,16) ! OR USE I1 OR I2=0 TO OMIT WRITE.     00009880
C                                                                 00009890
        SAVE                                                      00009900
        INTEGER*4 ABSVAL(4),INCRVAL(4)                            00009910
        CALL PROCINFO(ABSVAL,INCRVAL)                             00009920
        TIMES=SECNDS(TIME0)                                       00009930
        MIN=TIMES/60.0                                            00009940
        SEC=AMOD(TIMES,60.0)                                      00009950
        CPUSEC=INCRVAL(1)*.01                                     00009960
        IMIN=CPUSEC/60.0                                          00009970
        CSEC=AMOD(CPUSEC,60.0)                                    00009980
        PCPU=100.*(CPUSEC/TIMES)                                  00009990
        IF(I1.NE.0) THEN                                          00010000
          IF(I1.GT.0) THEN                                        00010010
            J=1                                                   00010020
          ELSE                                                    00010030
            J=0                                                   00010040
          ENDIF                                                   00010050
          WRITE(IABS(I1),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU, 00010060
       1 (INCRVAL(I),I=2,4)                                       00010070
60      FORMAT(I1,65('s')/' TOTAL "ELAPSED" TIME=',F16.2,' SEC. (', 00010080
       1 I4,' MIN.',F6.2,' SEC.)'/                               00010090
       2 ' CPU_TIME=',F15.2,' SEC. (',I4,' M. ',F5.2,            00010100
       1 ' S.)    CPU % =',F6.2,'%'/                             00010110
       3 ' BUF.I/O_COUNT=',I10/                                  00010120
       4 ' DIR.I/O_COUNT=',I10/                                  00010130
       5 ' PAGE_FAULTS=',2X,I10/                                 00010140
       6 ' ',65('s')//)                                          00010150
        ENDIF                                                     00010160
        IF(I2.NE.0) THEN                                          00010170
          IF(I2.GT.0) THEN                                        00010180
            J=1                                                   00010190
          ELSE                                                    00010200
            J=0                                                   00010210
          ENDIF                                                   00010220
          WRITE(IABS(I2),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU, 00010230
```

```
      1 (INCRVAL(I),I=2,4)                                     00010240
        ENDIF                                                  00010250
        RETURN                                                 00010260
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'  00010270
        ENTRY SETTIME()                                        00010280
        TIME0=SECNDS(0.0)                                      00010290
        CALL PROCINFO(ABSVAL,INCRVAL)                          00010300
        RETURN                                                 00010310
        END                                                    00010320
        SUBROUTINE DECODEIX(NUMFLD,NUMLEN,IX,*)                00010330
C--USED IN CALL NAMELIST(IUNIT,'SNAME',*)                      00010340
        CHARACTER*9 FMT                                        00010350
        CHARACTER*20 NUMFLD                                    00010360
        IF(NUMLEN.LT.1) RETURN 1                               00010370
        IDIFF=20-NUMLEN                                        00010380
        IF(IDIFF.EQ.0) THEN                                    00010390
          ENCODE(9,991,FMT) NUMLEN                             00010400
        ELSE                                                   00010410
          ENCODE(9,992,FMT) NUMLEN,IDIFF                       00010420
        ENDIF                                                  00010430
991     FORMAT('(I',I2,'      )')                              00010440
992     FORMAT('(I',I2,',',I2,'X)')                            00010450
        DECODE(9,FMT,NUMFLD) IX                                00010460
        RETURN                                                 00010470
        END                                                    00010480
        SUBROUTINE DECODEX(NUMFLD,NUMLEN,NDEC,X,*)             00010490
C--USED IN CALL NAMELIST(IUNIT,'SNAME',*)                      00010500
        CHARACTER*12 FMT                                       00010510
        CHARACTER*20 NUMFLD                                    00010520
        IF(NUMLEN.LT.1) RETURN 1                               00010530
        LENDEC=NUMLEN-NDEC                                     00010540
        IDIFF=20-NUMLEN                                        00010550
        IF(IDIFF.EQ.0) THEN                                    00010560
         ENCODE(12,991,FMT) NUMLEN,LENDEC                      00010570
        ELSE                                                   00010580
         ENCODE(12,992,FMT) NUMLEN,LENDEC,IDIFF                00010590
        ENDIF                                                  00010600
991     FORMAT('(F',I2,'.',I2,'      )')                       00010610
992     FORMAT('(F',I2,'.',I2,',',I2,'X)')                     00010620
        DECODE(12,FMT,NUMFLD) X                                00010630
        RETURN                                                 00010640
        END                                                    00010650
        REAL*8 FUNCTION DERF(X)                                00010660
C                                                              00010670
C   DERF COMPUTES THE ERROR FUNCTION TO ABOUT 15-PLACES.       00010680
C   SEE MATH. OF COMP., V.22,N.101,JAN,1968.                   00010690
C                                                              00010700
        IMPLICIT REAL*8 (A-H,O-Z)                              00010710
        DIMENSION A1(19),A2(19)                                00010720
        DATA PI/3.141592653589793D0/                          00010730
        DATA A1/.7032250027437754D0,.3305015219166062D0,      00010740
       1 .2013397472647063D0,.1086302450227407D0,             00010750
       2 .4677552343248486D-1,.15398572615571020D-1,          00010760
       3 .38015076798529987D-2,.69718379240802287D-3,         00010770
       4 .94490926881045500D-4,.94328116983836680D-5,         00010780
```

```
    5 .691927520325140lD-6,.37225234936910800-7,               00010790
    6 .1466606142338001D-8,.4226161443180490-10,               00010800
    7 .8897865267233D-12,.136760444757D-13,                     00010810
    8 .15334234250-15,.12536751D-17,.74517D-20/                 00010820
      DATA A2/.2472551681400521D0,.1442272263615747D0,          00010830
    1 .8698945499593455D-1,.43977338194083370-1,               00010840
    2 .1724396258866226D-1,.5079069612202570D-2,               00010850
    3 .1108606453423407D-2,.1782280162548617D-3,               00010860
    4 .2104045830732514D-4,.1820663163643408D-5,               00010870
    5 .1153309909443694D-6,.5342750276030827D-8,               00010880
    6 .1808485878095127D-9,.44696822924881D-11,                 00010890
    7 .8060688389450-13,.106013364636D-14,                      00010900
    8 .101649277D-16,.710005D-19,0.0D0/                         00010910
      IF(X.EQ.0.0D0) THEN                                       00010920
        DERF=0.0D0                                              00010930
        RETURN                                                  00010940
      ENDIF                                                     00010950
        B=.4D0*X                                                00010960
        S=DSIN(B)                                               00010970
        C=DCOS(B)                                               00010980
        C2=C+C                                                  00010990
        ALP=C2*C-1.D0                                           00011000
        SUM=0.0D0                                               00011010
        DO 10 N=1,19                                            00011020
          SUM=SUM+(A1(N)+C2*A2(N))*ALP**(N-1)                   00011030
   10     CONTINUE                                              00011040
      DERF=B/PI+S*SUM                                           00011050
      RETURN                                                    00011060
      END                                                       00011070
      SUBROUTINE DFIND(X0,NHALF,XM,FCT,XL,XR,IER)               00011080
C                                                               00011090
C--"FIND" FIRST FCT(XL) AND FCT(XR) WITH OPPOSITE SIGNS         00011100
C  IN RANGE X=(X0,XM) USING MAX. 2*NHALF BISECTIONS, WHERE      00011110
C  FCT IS A REAL*8 EXTERNAL DECLARED FUNCTION.                  00011120
C                                                               00011130
C--IF NO SIGN CHANGE IN (X0,XM), EXITS WITH IER=1               00011140
C  (ELSE IER=0 MEANS XL AND XR FOUND).                          00011150
C                                                               00011160
C--USE BEFORE CALL DRTMI(X,F,FCT,XL,XR,EPS,MAXITR,IER)          00011170
C  TO FIND GUARANTEED ROOT OF FCT(X)=0 BY MUELLERS ITERATION.   00011180
C                                                               00011190
      IMPLICIT REAL*8 (A-H,O-Z)                                 00011200
      XL=X0                                                     00011210
      SIGNXR=DSIGN(1.D0,FCT(XM))                                00011220
      XR=XM                                                     00011230
      DO N=1,NHALF                                              00011240
        X=0.5D0*(XL+XR)                                         00011250
        IF(DSIGN(1.D0,FCT(X)).NE.SIGNXR) GO TO 10              00011260
        XR=X                                                    00011270
      ENDDO                                                     00011280
      XL=0.5D0*(X0+XM)                                          00011290
      XR=XM                                                     00011300
      DO N=1,NHALF                                              00011310
        X=0.5D0*(XL+XR)                                         00011320
        IF(DSIGN(1.D0,FCT(X)).NE.SIGNXR) GO TO 20              00011330
```

```
          XL=X                                                      00011340
     ENDDO                                                          00011350
     IER=1                                                          00011360
     RETURN                                                         00011370
10   IER=0                                                          00011380
     XL=X                                                           00011390
     RETURN                                                         00011400
20   IER=0                                                          00011410
     XR=X                                                           00011420
     END                                                            00011430
     SUBROUTINE DRTMI(X,F,FCT,XLI,XRI,EPS,IEND,IER)                 00011440
C--IBM SSP ROUTINE, P.219, TO SOLVE FCT(X)=0 BY MUELLERS ITERATION  00011450
     IMPLICIT REAL*8 (A-H,O-Z)                                      00011460
     IER=0                                                          00011470
     XL=XLI                                                         00011480
     XR=XRI                                                         00011490
     X=XL                                                           00011500
     TOL=X                                                          00011510
     F=FCT(TOL)                                                     00011520
     IF(F)1,16,1                                                    00011530
1    FL=F                                                           00011540
     X=XR                                                           00011550
     TOL=X                                                          00011560
     F=FCT(TOL)                                                     00011570
     IF(F)2,16,2                                                    00011580
2    FR=F                                                           00011590
     IF(DSIGN(1.D0,FL)+DSIGN(1.D0,FR))25,3,25                       00011600
3    I=0                                                            00011610
     TOLF=100.D0*EPS                                                00011620
4    I=I+1                                                          00011630
     DO 13 K=1,IEND                                                 00011640
     X=.5D0*(XL+XR)                                                 00011650
     TOL=X                                                          00011660
     F=FCT(TOL)                                                     00011670
     IF(F)5,16,5                                                    00011680
5    IF(DSIGN(1.D0,F)+DSIGN(1.D0,FR))7,6,7                          00011690
6    TOL=XL                                                         00011700
     XL=XR                                                          00011710
     XR=TOL                                                         00011720
     TOL=FL                                                         00011730
     FL=FR                                                          00011740
     FR=TOL                                                         00011750
7    TOL=F-FL                                                       00011760
     A=F*TOL                                                        00011770
     A=A+A                                                          00011780
     IF(A-FR*(FR-FL))8,9,9                                          00011790
8    IF(I-IEND)17,17,9                                              00011800
9    XR=X                                                           00011810
     FR=F                                                           00011820
     TOL=EPS                                                        00011830
     A=DABS(XR)                                                     00011840
     IF(A-1.D0)11,11,10                                             00011850
10   TOL=TOL*A                                                      00011860
11   IF(DABS(XR-XL)-TOL)12,12,13                                    00011870
12   IF(DABS(FR-FL)-TOLF)14,14,13                                   00011880
```

```
13      CONTINUE                                              00011890
        IER=1                                                 00011900
14      IF(DABS(FR)-DABS(FL))16,16,15                         00011910
15      X=XL                                                  00011920
        F=FL                                                  00011930
16      RETURN                                                00011940
17      A=FR-F                                                00011950
        DX=(X-XL)*FL*(1.D0+F*(A-TOL)/(A*(FR-FL)))/TOL         00011960
        XM=X                                                  00011970
        FM=F                                                  00011980
        X=XL-DX                                               00011990
        TOL=X                                                 00012000
        F=FCT(TOL)                                            00012010
        IF(F)18,16,18                                         00012020
18      TOL=EPS                                               00012030
        A=DABS(X)                                             00012040
        IF(A-1.D0)20,20,19                                    00012050
19      TOL=TOL*A                                             00012060
20      IF(DABS(DX)-TOL)21,21,22                              00012070
21      IF(DABS(F)-TOLF)16,16,22                              00012080
22      IF(DSIGN(1.D0,F)+DSIGN(1.D0,FL))24,23,24             00012090
23      XR=X                                                  00012100
        FR=F                                                  00012110
        GO TO 4                                               00012120
24      XL=X                                                  00012130
        FL=F                                                  00012140
        XR=XM                                                 00012150
        FR=FM                                                 00012160
        GO TO 4                                               00012170
25      IER=2                                                 00012180
        RETURN                                                00012190
        END                                                   00012200
        SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)            00012210
C                                                             00012220
C    GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780      00012230
C                                                             00012240
C    MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                      00012250
C    ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2  00012260
C            > 0 FOR ONE BLANK LINE BEFORE.                   00012270
C    IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).  00012280
C    IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).  00012290
C                                                             00012300
C    MESSAGES ARE WRITTEN IN THE FORM:                        00012310
C                                                             00012320
C    (ERRMSG): _MSG_HERE_                                     00012330
C                                                             00012340
        CHARACTER*(*) MSG                                     00012350
        I=LEN(MSG)                                            00012360
        DO 1 J=1,2                                            00012370
          IF(J.EQ.1) THEN                                     00012380
        JUNIT=IUNIT1                                          00012390
            ELSE                                              00012400
        JUNIT=IUNIT2                                          00012410
          ENDIF                                               00012420
          IF(JUNIT.GT.0) THEN                                 00012430
```

```
          IF(ISKIP.EQ.0) THEN                                      00012440
            WRITE(JUNIT,2) MSG                                     00012450
            ELSE                                                   00012460
        WRITE(JUNIT,3) MSG                                         00012470
          ENDIF                                                    00012480
        ENDIF                                                      00012490
1       CONTINUE                                                   00012500
        CALL EXIT                                                  00012510
2       FORMAT(1X,'{ERRMSG}: ',A<I>)                              00012520
3       FORMAT(/1X,'{ERRMSG}: ',A<I>)                             00012530
        END                                                        00012540
        SUBROUTINE INTEG1(N,X,Y,Y0)                                00012550
C          THIS ROUTINE INTEGRATES A FUNCTION'S VALUES (Y          00012560
C          AS A FUNCTION OF X) FROM 0 TO X BY CALCULATING THE CUBIC 00012570
C          SPLINE COEFFICIENTS AND INTEGRATING THE RESULTING       00012580
C          CUBIC POLYNOMIAL APPROXIMATION.  THE Y VALUES ARE       00012590
C          REPLACED BY THE INTEGRATED VALUES.                      00012600
C          Y0 IS THE VALUE OF Y AT X=0.0 (ASSUMES THAT ALL INPUT   00012610
C          X > 0).                                                 00012620
        DIMENSION X(N),Y(N)                                        00012630
        DIMENSION A(200),B(200),C(200),P(200),S(200),PS(2),X1(200),Y1(200)00012640
        DATA PS/0.0,0.0/                                           00012650
        DO 1 I=1,N                                                 00012660
        X1(I+1)=X(I)                                               00012670
      1 Y1(I+1)=Y(I)                                               00012680
        X1(1)=0.0                                                  00012690
        Y1(1)=Y0                                                   00012700
        N1=N+1                                                     00012710
        CALL SPLIN1(N1,0,X1,Y1,A,B,C,0,PS,P,S)                     00012720
        Y(1)=X(1)*(Y0+X(1)*A(1)/2.+X(1)*X(1)*B(1)/3.+X(1)**3*C(1)/4.) 00012730
        N1=N-1                                                     00012740
        DO 10 I=1,N1                                               00012750
        Z=X(I+1)-X(I)                                              00012760
     10 Y(I+1)=Y(I)+Z*(Y1(I+1)+A(I+1)*Z/2.+B(I+1)*Z*Z/3.+C(I+1)*Z**3/4.) 00012770
        RETURN                                                     00012780
        END                                                        00012790
        SUBROUTINE MINMAX(A,N,AMIN,AMAX)                           00012800
        DIMENSION A(1)                                             00012810
        AMIN=A(1)                                                  00012820
        AMAX=AMIN                                                  00012830
        DO 1 I=2,N                                                 00012840
        AMIN=AMIN1(AMIN,A(I))                                      00012850
        AMAX=AMAX1(AMAX,A(I))                                      00012860
      1 CONTINUE                                                   00012870
        RETURN                                                     00012880
        END                                                        00012890
        SUBROUTINE NLSOL(FCODE,PCODE,SUBZ,SUBEND)                  00012900
C                                                                  00012910
C {NLSOL}: GENERAL NONLINEAR LEAST-SQUARES SOLUTION    {2/8/82}    00012920
C          USING DENNIS ET AL (1979; SEE REF1 BELOW)               00012930
C          ADAPTIVE NONLINEAR LEAST-SQUARES ALGORITHM.             00012940
C                                                                  00012950
C** THIS IS AN INTERFACE ROUTINE WRITTEN FOR THE VAX-11/780 BY     00012960
C   W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.         00012970
C                                                                  00012980
```

```
C** THIS INTERFACE (NLSOL) HAS ADDITIONAL OPTIONS (BESIDE REF1) TO:    00012990
C    (1) PERFORM EITHER UNCONSTRAINED OR UP TO 4-TYPES OF CONSTRAINED   00013000
C        ADAPTIVE NONLINEAR REGRESSION FOR ARBITRARY NONLINEAR PROBLEMS. 00013010
C        (I.E., PARTIAL OR FULL LOWER/HIGHER PARAMETER BOUNDS, ETC.)    00013020
C    (2) HOLDING CERTAIN PARAMETERS FIXED (I.E., AS CONSTANTS) IN THE   00013030
C        LEAST-SQUARES (THIS IS ANOTHER FORM OF CONSTRAINING SOLUTION   00013040
C        SPACE).                                                        00013050
C    (3) PROVIDE FOR WEIGHTED OBSERVATIONS (I.E., WEIGHTED LEAST-SQUARES)00013060
C    (4) OBJECT (RUN)-TIME CONTROL OF READING THE DATA MATRIX, PLUS     00013070
C        MANY OTHER I/O OPTIONS, ETC.                                   00013080
C    (5) OPTIONALLY, ONE CAN USE EITHER ESTIMATED PARTIAL DERIVATIVES, OR00013090
C        ANALYTICAL PARTIAL DERIVATIVES (IF SUBROUTINE PCODE AVAILABLE). 00013100
C                                                                       00013110
C** THE USER ONLY NEEDS TO WRITE SUBROUTINES FCODE, PCODE, SUBZ, AND    00013120
C    SUBEND (SEE DETAILS BELOW) EXACTLY AS USED IN SUBROUTINE 'MARQRT'  00013130
C    (SEE REF2) OR 'IMSLMQ' (SEE REF3).  ALSO, THE SAME PARAMETER FILE  00013140
C    FOR005 AND OBJECT (RUN)-TIME DATA MATRIX FILE FOR010 AS USED BY    00013150
C    EITHER MARQRT OR IMSLMQ MAY BE USED IN 'NLSOL'.                    00013160
C                                                                       00013170
C** NLSOL CALLS NLITR WHICH CALLS 'NL2ITR' AS PUBLISHED BY DENNIS ET AL,00013180
C    (SEE REF1, P. 38), OR 'NL2SNO' (SEE REF1, P. 35).                  00013190
C                                                                       00013200
C** REF1:  DENNIS, J.E., ET AL, 1979, AN ADAPTIVE NONLINEAR LEAST-      00013210
C          SQUARES ALGORITHM, NTIS REPORT AD-A079-716.                 00013220
C                                                                       00013230
C    REF2:  ANDERSON, W.L., 1980, PROGRAM MARQHXY: INVERSION OF HX AND HY00013240
C           FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN- 00013250
C           FILE REPT. 80-901.                                         00013260
C                                                                       00013270
C    REF3:  ANDERSON, W.L., 1980, PROGRAM IMSLEXY: INVERSION OF EX AND EY00013280
C           FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN- 00013290
C           FILE REPT. 80-1073.                                        00013300
C                                                                       00013310
C***********************************************************************00013320
C                                                                       00013330
C****   THE USER MUST DECLARE THE CALLING PARAMETERS AS EXTERNAL IN THE 00013340
C       CALLING PROGRAM (ANY DESIRED NAMES MAY BE USED).               00013350
C E.G.,                                                                 00013360
C                                                                       00013370
C (MAIN):                                                               00013380
C    EXTERNAL MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND                       00013390
C    CALL NLSOL(MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND)                    00013400
C    STOP    !<OR USE>: CALL EXIT                                       00013410
C    END                                                               00013420
C [FCODE]:                                                              00013430
C    SUBROUTINE MY_FCODE(Y,X,B,W,F,IN,IDER)                            00013440
C    USER WRITTEN TO EVALUATE THE NONLINEAR OBJECTIVE FUNCTION (F)      00013450
C    USED IN NLSOL AS THE WEIGHTED SUM OF (Y(IN)-F)**2, WHERE           00013460
C    Y= OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N, WHERE N IS           00013470
C      GIVEN IN $PARMS NAMELIST INPUT--SEE BELOW).                      00013480
C    X= OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,M, WHERE            00013490
C    M IS IN $PARMS INPUT).                                            00013500
C    B= CURRENT PARAMETER ESTIMATES (DIM. K, WHERE                      00013510
C    K IS IN $PARMS INPUT).                                            00013520
C    W= WORK ARRAY (DIM. 5)--MAY BE USED TO PASS DATA TO PCODE.         00013530
```

```
C      F= (OUTPUT) THE FUNCTION VALUE EVALUATED FOR THE GIVEN         00013540
C      Y,X, AND B ARRAYS AT THE OBSERVATION NO. 'IN'.                 00013550
C      IN= (INPUT) OBSERVATION NO. TO EVALUATE F (1.LE.IN.LE.N),      00013560
C      WHICH IS CONTROLLED EXTERNALLY BY 'NLSOL'. USUALLY,            00013570
C      IN=1,2,....,N--BUT NOT ALWAYS.                                 00013580
C      IDER= 0 IF ANALYTICAL DERIVATIVES ARE USED (PCODE CALLED       00013590
C         AFTER FCODE).                                               00013600
C         = 1 IF ESTIMATED DERIVATIVES ARE USED (PCODE NOT CALLED     00013610
C         AFTER FCODE).                                               00013620
C      DIMENSION Y(1),X(500,5),B(1),W(5)                              00013630
C>>>>> INSERT USER CODE HERE TO EVALUATE F  <<<<<                     00013640
C      END                                                            00013650
C [PCODE]: >> PCODE MAY BE A DUMMY NAME IF ONLY IDER=1 IS TO BE USED. <<00013660
C      SUBROUTINE MY_PCODE(P,X,B,W,F,IN,IP,IB)                        00013670
C      USER WRITTEN TO EVALUATE THE ANALYTICAL PARTIAL DERIVATIVES OF 00013680
C      F WITH RESPECT TO B(J),J=1,2,....,K, AT OBSERVATION 'IN', WHERE00013690
C      P= (OUTPUT) PARTIAL DERIVATIVE ARRAY (DIM. K, WHERE            00013700
C      K IS IN SPARMS INPUT).                                         00013710
C      X,B,W ARE THE SAME AS USED IN FCODE (SEE ABOVE).               00013720
C      F= LAST FUNCTION VALUE FROM FCODE AT OBSERVATION IN.           00013730
C      (NOTE THAT F MAY NOT BE NEEDED, BUT IS AVAILABLE ANYWAY)       00013740
C      IN= (INPUT) OBSERVATION NO. TO EVALUATE P ARRAY, WHICH IS      00013750
C      CONTROLLED EXTERNALLY BY 'NLSOL' (1.LE.IN.LE.N).               00013760
C      IP= (INPUT) THE NO. OF B-PARAMETERS HELD FIXED IN THE LEAST-   00013770
C      SQUARES (0.LE.IP.LE.K-1; USE IP=0 IF NONE).                    00013780
C      IB= ARRAY OF B-PARAMETER INDICES HELD FIXED IF IP.GT.0.        00013790
C      NOTE THAT THE INDICES IN IB ARRAY MAY BE IN ANY ORDER,         00013800
C      BUT MUST BE BETWEEN 1 AND K (K IS IN SPARMS INPUT).            00013810
C      DIMENSION P(1),X(500,5),B(1),W(5),IB(1)                        00013820
C>>>>>  INSERT USER CODE HERE TO EVALUATE P  <<<<<                    00013830
C      END                                                            00013840
C [SUBZ]:                                                             00013850
C      SUBROUTINE MY_SUBZ(Y,X,B,W,NW,N,TITLE,IOUT)                    00013860
C      USER WRITTEN INITIALIZATION ROUTINE (CALLED ONCE BY 'NLSOL').  00013870
C      SUBZ MAY BE USED TO CHECK Y(IN),X(IN,M) AFTER INPUT VIA        00013880
C      OBJECT (RUN)-TIME INPUT (SEE BELOW) ON UNIT IALT. ALSO, SUBZ   00013890
C      MAY BE USED TO READ ADDITIONAL SINIT PARAMETERS, AND TO LOAD   00013900
C      ANY COMMON BLOCKS IF NEEDED IN THE USERS FCODE,PCODE.          00013910
C      Y,X,B,W  ARE THE SAME AS USED IN FCODE (SEE ABOVE).            00013920
C      NW= USE ANY DUMMY INTEGER VARIABLE (THIS IS                    00013930
C      TO MAINTAIN COMPATIBILITY WITH 'MARQRT' OR 'IMSLMQ').          00013940
C      N= NO. OF OBSERVATIONS IN Y(N),X(N,M) ARRAYS, WHERE            00013950
C      K.GE.N.LE.500 (N,M,K  ARE IN SPARMS INPUT).                    00013960
C      TITLE= (INPUT) 80-CHARACTER HEADING (SEE INPUT FOR005 BELOW).  00013970
C      IOUT= 1 IF TO WRITE OUTPUT ON BOTH FOR006 AND FOR016.          00013980
C         = 0 IF TO WRITE OUTPUT ONLY ON FOR006.                      00013990
C      DIMENSION Y(1),X(500,5),B(1),W(5)                              00014000
C      CHARACTER*80 TITLE                                             00014010
C>>>>> INSERT USER CODE HERE FOR ANY INITIALIZATION DESIRED  <<<<<    00014020
C      END                                                            00014030
C [SUBEND]:                                                           00014040
C      SUBROUTINE MY_SUBEND(Y,X,B,K,N,TITLE,IOUT)                     00014050
C      USER WRITTEN TERMINATION ROUTINE (CALLED ONCE BY 'NLSOL').     00014060
C      SUBEND MAY BE USED TO OUTPUT THE FINAL SOLUTION VECTOR B(I),   00014070
C      I=1,2,....,K, IN OTHER FORMS, ETC., AS DESIRED. [OR IT MAY BE A 00014080
```

```
C      DUMMY ROUTINE; I.E., JUST RETURNS.]                          00014090
C      Y,X,K,N,TITLE,IOUT  ARE THE SAME AS IN SUBZ AND FCODE.       00014100
C      B= (INPUT) IS THE FINAL SOLUTION VECTOR AS DETERMINED BY     00014110
C       'NLSOL' (SEE REF1 FOR DETAILS).                             00014120
C      DIMENSION Y(1),X(500,5),B(1)                                 00014130
C      CHARACTER*80 TITLE                                           00014140
C>>>>>  INSERT USER CODE HERE FOR ANY TERMINATION SUMMARY DESIRED  <<<<<00014150
C      END                                                          00014160
C                                                                   00014170
C**********************************************************************00014180
C                                                                   00014190
C** INPUT ORDER ON FOR005 (PARAMETER FILE LOGICAL NAME):            00014200
C                                                                   00014210
C 1.   TITLE (MAX. 80-CHARACTERS--ALWAYS READ BEFORE $PARMS INPUT). 00014220
C 2.   $PARMS (SAME DEFINITIONS AS IN 'MARQRT', REF2, OR IN 'IMSLMQ',00014230
C      REF3), WHICH INCLUDES: N,K,IP,M,IALT,ISTOP,IWT,IDER,         00014240
C      IPRT,NITER,IOUT,SP,B(),IB(); PLUS THE FOLLOWING PARAMETERS FROM00014250
C      REF1 (NL2SOL), P.31-35: IV(),V();  IN ADDITION, THE LOWER AND00014260
C      UPPER BOUND ARRAYS BL(),BH(), RESPECTIVELY, ARE REQUIRED IF  00014270
C      SP>2.                                                        00014280
C 3.   (OBJECT-RUN-TIME FORMAT STATEMENT) TO DESCRIBE THE FORMAT OF THE00014290
C      DATA MATRIX ROW Y(I),(X(I,J),J=1,M*) READ ON FILE IALT, WHERE00014300
C      M*=M (IF IWT=0) OR M*=M+1 (IF IWT>0), M.LE.4, AND I=1,2,....,N.00014310
C (3A). INSERT DATA MATRIX HERE ONLY IF IALT=5.                     00014320
C 4.   $INIT OPTIONAL NAMELIST USED FOR READING PROBLEM-DEPENDENT   00014330
C      PARAMETERS USED IN SUBROUTINE SUBZ (SEE ABOVE).  CURRENTLY,  00014340
C      THE FOLLOWING $INIT NAMES (AND DIM.) CAN BE USED: IOB,MM,XO,YO,00014350
C      L,EP,EPS,NEPS,METHOD,NFIN,IER,MEV,IOPT,NSIG,MAXFN,DELTA,PARM(4),00014360
C      AND IRATIO(2).                                               00014370
C 5.   OPTIONALLY, REPEAT STEPS 1-4, IF PARAMETER ISTOP=0 WAS USED  00014380
C      IN THE LAST STEP 2.                                          00014390
C                                                                   00014400
C** OUTPUT IS GIVEN ON FOR006 (ON-LINE USUALLY) AND ON FOR016(IF IOUT=1)00014410
C    FOR016 CONTAINS ALL PRINTABLE OUTPUT SELECTED VIA $PARMS IPRT,IOUT.00014420
C    NOTE: IPRT=0 GIVES ABBREVIATED OUTPUT ON FOR006 (BUT MORE ON FOR016)00014430
C          IPRT=1 OR -2 GIVES DETAILED OUTPUT ON BOTH 6 AND 16.     00014440
C          IPRT=-1 GIVES MODERATE OUTPUT ON 6 (DETAILED ON 16).     00014450
C                                                                   00014460
C** TO RUN ON VAX (ELIMINATE <> DELIMITERS IN SUBSTITUTIONS):       00014470
C                                                                   00014480
C    $ASSIGN <PARAMETER FILE NAME> FOR005                           00014490
C    $ASSIGN <DATA MATRIX FILE NAME> FOR010                         00014500
C    $RUN <MAIN NAME>                                               00014510
C                                                                   00014520
C**********************************************************************00014530
C                                                                   00014540
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00014550
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF     00014560
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                    00014570
C      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                          00014580
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.00014590
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:        00014600
C      PARAMETER (K1DIM=KDIM-1,K2DIM=KDIM+KDIM,M1DIM=MDIM-1,        00014610
C     1 IVDIM=KDIM+60,NKVDIM=96+2*NDIM+(KDIM*(7*KDIM+41))/2)        00014620
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00014630
```

```
      C                                                             00014640
            REAL*4 L                                                00014650
            DIMENSION B(KDIM),SQWT(NDIM),IB(K1DIM),C(KDIM),INDEX(KDIM),  00014660
           1 IV(IVDIM),V(NKVDIM),CBOUND(K2DIM),                    00014670
           2 BL(KDIM),BH(KDIM),CL(KDIM),CH(KDIM),SE(KDIM),         00014680
           3 W(KDIM),PARM(4),IRATIO(2),PRNT(5)                     00014690
            INTEGER SP,SCALEP,SY,SCALEY                            00014700
            CHARACTER*3 CHAR3                                      00014710
            CHARACTER*6 CALLED                                     00014720
            CHARACTER*80 TITLE                                     00014730
            CHARACTER*132 LINE132                                  00014740
            CHARACTER*72 FMT                                       00014750
            COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,  00014760
           1 IDER_,K_,ISP                                          00014770
            COMMON/BOUNDS/BL_(KDIM),BH_(KDIM)                      00014780
            COMMON/REVCOM/R(NDIM)                                  00014790
            EQUIVALENCE (SQWT(1),X(1,MDIM)),(N,NOBS),(K,KPARMS),(M,MVARS),  00014800
           1 (CL(1),CBOUND(1)),(CH(1),CBOUND(KDIM+1))              00014810
            EXTERNAL FCODE,PCODE,CALCR                             00014820
      C**                                                          00014830
      C  THE FOLLOWING COMMON/NAME_LIST/ IS TO SIMULATE ON VAX-11/780:  00014840
      C     NAMELIST/PARMS/ & READ(5,PARMS) VIA 'CALL NAMELIST(5,'SPARMS',*)'  00014850
      C     NAMELIST/INIT/ & READ(5,INIT)  VIA 'CALL NAMELIST(5,'SINIT',*)'  00014860
      C** SEE SUBROUTINE NAMELIST FOR MORE DETAILS, AND ALSO REF1-REF3 FOR  00014870
      C  DETAILS ON EACH PARAMETER DEFINITION.                     00014880
      C**                                                          00014890
            COMMON/NAME_LIST/N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,INON,  00014900
           1 FF,T,E,TAU,XL,MODLAM,GAMCR,DEL,ZETA,IOUT,SP,SCALEP,SY,SCALEY,  00014910
           2 B,IB, IOB,MM,XO,YO,L,EP,EPS,NEPS,METHOD,NFIN,IER,MEV,  00014920
           3 IV,V,BL,BH,                                           00014930
           4 IOPT,NSIG,MAXFN,DELTA,PARM, H,IRATIO                  00014940
      C**                                                          00014950
      C  NOTE THAT COMMON/NAME_LIST/ CONTAINS SOME PARAMETERS ONLY FOR  00014960
      C  COMPATIBILITY WITH 'MARQRT' OR 'IMSLMQ'; I.E., THE FOLLOWING LIST  00014970
      C  OF PARAMETERS ARE CURRENTLY NOT USED DIRECTLY BY 'NLSOL':  00014980
      C     INON,FF,T,TAU,XL,MODLAM,GAMCR,DEL,E,ZETA,SY,SCALEY,SCALEP,  00014990
      C     IOPT,NSIG,MAXFN,DELTA,PARM.                            00015000
      C**                                                          00015010
      C                                                            00015020
      C** READ NLSOL TITLE LINE                                    00015030
            READ(5,10,ERR=9000,END=9010) TITLE                     00015040
      10    FORMAT(A80)                                            00015050
      C                                                            00015060
      C**PRESET DEFAULT PARMS (SOME MUST BE GIVEN IN SPARMS ELSE AN ERROR)  00015070
      C                                                            00015080
            N=0                                                    00015090
            K=0                                                    00015100
            IP=0                                                   00015110
            M=0                                                    00015120
            IALT=10                                                00015130
            ISTOP=1                                                00015140
            ICALL=1                                                00015150
            IWT=0                                                  00015160
            IDER=0                                                 00015170
            IPRT=0                                                 00015180
```

```
      NITER=10                                                    00015190
      IOUT=1                                                      00015200
      SP=0                                                        00015210
      DO 20 I=1,KDIM                                              00015220
      IF(I.LT.KDIM) IB(I)=0                                       00015230
      BL(I)=0.0                                                   00015240
      B(I)=0.0                                                    00015250
      BH(I)=0.0                                                   00015260
20    CONTINUE                                                    00015270
22    IV(1)=10                                                    00015280
C**                                                               00015290
C  PRESET NLITR                                                   00015300
C**                                                               00015310
      CALL DFAULT(IV,V)                                           00015320
C**                                                               00015330
C** OVERRIDE FOR IV(15)=3 DEFAULT (MAY BE CHANGED VIA SPARMS INPUT) 00015340
C**                                                               00015350
      IV(15)=3                                                    00015360
C**                                                               00015370
C  READ SPARMS ON FOR005 VIA 'CALL NAMELIST' ON VAX              00015380
C**                                                               00015390
30    CALL NAMELIST(5,'SPARMS',*9020)                            00015400
C**                                                               00015410
C  SET EQUIVALENT PARAMETERS IN DIFFERENT COMMON'S               00015420
C**                                                               00015430
      ISP=SP                                                      00015440
      DO 32 I=1,KDIM                                              00015450
      BFIX(I)=B(I)                                                00015460
      BL_(I)=BL(I)                                                00015470
      BH_(I)=BH(I)                                                00015480
      IF(I.LT.KDIM) IIB(I)=IB(I)                                  00015490
32    CONTINUE                                                    00015500
      IIP=IP                                                      00015510
      IDER_=IDER                                                  00015520
      K_=K                                                        00015530
C**                                                               00015540
C  TEST SPARMS BEFORE PROCEEDING                                 00015550
C**                                                               00015560
      IF(IP.LT.0.OR.IP.GT.K1DIM)CALL ERRMSG('IP<0 OR IP>19',0,6,16) 00015570
      KIP=K-IP                                                    00015580
      IF(N.LT.1.OR.N.GT.NDIM.OR.N.LT.KIP)                        00015590
     1 CALL ERRMSG('N<1,N>500,OR N<K-IP',0,6,16)                 00015600
      IF(K.LT.1.OR.K.GT.KDIM.OR.KIP.LT.1)                        00015610
     1 CALL ERRMSG('K<1,K>20,OR K-IP<1',0,6,16)                  00015620
      IF(M.LT.1.OR.M.GT.M1DIM)CALL ERRMSG('M<1 OR M>4',0,6,16)   00015630
      IF(IALT.EQ.6.OR.IALT.EQ.13.OR.IALT.EQ.16.OR.IALT.EQ.4)     00015640
     1 CALL ERRMSG('IALT=4,6,13,OR 16',0,6,16)                   00015650
      IF(ISTOP.EQ.0.AND.IALT.EQ.5)                               00015660
     1    CALL ERRMSG('ISTOP=0 BUT IALT=5',0,6,16)               00015670
      IF(IWT.LT.0.OR.IWT.GT.2)CALL ERRMSG('IWT<0 OR IWT>2',0,6,16) 00015680
      IF(IDER.LT.0.OR.IDER.GT.1)CALL ERRMSG('IDER<0 OR IDER>1',0,6,16) 00015690
      IF(SP.LT.0.OR.SP.GT.4)CALL ERRMSG('SP<0 OR SP>4',0,6,16)   00015700
      IF(IP.GT.0) THEN                                           00015710
        DO J=1,IP                                                00015720
          IF(IB(J).LT.1.OR.IB(J).GT.K) THEN                      00015730
```

```
            ENCODE(3,43,CHAR3) J                                      00015740
            CALL ERRMSG('IP>0 AND IB(J)<1 OR IB(J)>K FOR J='//        00015750
    1       CHAR3,0,6,16)                                             00015760
          ENDIF                                                       00015770
        ENDDO                                                         00015780
      ENDIF                                                           00015790
      IF(SP.EQ.0.OR.SP.EQ.2) GO TO 41                                 00015800
      DO 40 I=1,KPARMS                                                00015810
        IF(SP.EQ.1) THEN                                             00015820
          IF(IP.GT.0) THEN                                           00015830
          DO 42 J=1,IP                                                00015840
            IF(I.EQ.IB(J)) GO TO 40                                   00015850
42        CONTINUE                                                    00015860
          ENDIF                                                       00015870
        IF(B(I).LE.0.) THEN                                           00015880
          ENCODE(3,43,CHAR3) I                                        00015890
43          FORMAT(I2,'.')                                            00015900
              CALL ERRMSG('SP=1 AND B(I)<=0 FOR I='//CHAR3,0,6,16)    00015910
            ENDIF                                                     00015920
        ELSE IF(SP.GT.2) THEN                                         00015930
          IF(B(I).LT.BL(I).OR.B(I).GT.BH(I).OR.BL(I).GT.BH(I)) THEN   00015940
          ENCODE(3,43,CHAR3) I                                        00015950
            CALL ERRMSG('SP>2 AND B(I)<BL(I), '//                     00015960
    1       'B(I)>BH(I), OR BL(I)>BH(I)'//                            00015970
    2       ' FOR I='//CHAR3,0,6,16)                                  00015980
          ENDIF                                                       00015990
        IF(BL(I).EQ.BH(I)) THEN                                       00016000
          IF(IP.GT.0) THEN                                            00016010
            DO 45 J=1,IP                                              00016020
              IF(I.EQ.IB(J)) GO TO 40                                 00016030
45          CONTINUE                                                  00016040
            ENDIF                                                     00016050
          ENCODE(3,43,CHAR3) I                                        00016060
          CALL ERRMSG('SP>2 AND BL(I)=BH(I) BUT B(I) NOT HELD '//     00016070
    1     'FIXED FOR I='//CHAR3,0,6,16)                               00016080
          ENDIF                                                       00016090
        ENDIF                                                         00016100
40      CONTINUE                                                      00016110
41      IF(IV(1).EQ.10) THEN                                          00016120
C**                                                                   00016130
C  NOTE CALL DFAULT(IV,V) WAS PRESET BEFORE SPARMS READ              00016140
C**                                                                   00016150
        IV(18)=NITER                                                  00016160
        IF(IPRT.GT.-3.AND.IPRT.LT.1) THEN                            00016170
          IV(19)=-1                                                   00016180
        ELSE                                                          00016190
          IV(19)=IPRT                                                 00016200
        ENDIF                                                         00016210
        IF(IOUT.EQ.0) THEN                                            00016220
          IV(21)=6                                                    00016230
        ELSE                                                          00016240
          IV(21)=16                                                   00016250
        ENDIF                                                         00016260
      ENDIF                                                           00016270
      IF(IP.GT.0) THEN                                                00016280
```

```
        DO 50 I=1,IP                                                00016290
          IF(IB(I).LE.0)CALL ERRMSG('IP>0 BUT SOME IB(I)<=0',0,6,16) 00016300
50      CONTINUE                                                    00016310
        ENDIF                                                       00016320
C                                                                   00016330
C   READ OBJECT(RUN)-TIME FORMAT FOR DATA MATRIX FROM FILE IALT.    00016340
C                                                                   00016350
        READ(5,60,ERR=9000,END=9010) FMT                           00016360
60      FORMAT(A72)                                                 00016370
        IF(IWT.EQ.0) THEN                                           00016380
          M1=MVARS                                                  00016390
        ELSE                                                        00016400
          M1=MVARS+1                                                00016410
        ENDIF                                                       00016420
        DO 70 I=1,NOBS                                              00016430
          READ(IALT,FMT,ERR=9030,END=9040) Y(I),(X(I,J),J=1,M1)    00016440
          IF(IWT.EQ.0.OR.X(I,M1).EQ.0.0) THEN                      00016450
            SQWT(I)=1.0                                             00016460
            GO TO 70                                                00016470
          ELSE IF(IWT.EQ.1) THEN                                    00016480
            SQWT(I)=1.0/X(I,M1)                                     00016490
          ELSE                                                      00016500
            SQWT(I)=1.0/SQRT(ABS(X(I,M1)))                          00016510
          ENDIF                                                     00016520
70      CONTINUE                                                    00016530
C                                                                   00016540
C   INITIALIZE VIA CALL SUBZ (READ SINIT AND TEST, LOAD COMMON, ETC.) 00016550
C                                                                   00016560
        CALL SUBZ(Y,X,BFIX,PRNT,NPRNT,N,TITLE,IOUT)                00016570
C       ***************************************************         00016580
C                                                                   00016590
C   WRITE SPARMS ON FOR006 AND FOR016 (THE LATTER IF IOUT=1)        00016600
C                                                                   00016610
        CALL NONBLANK(TITLE,NB)                                     00016620
        WRITE(6,80) TITLE,N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,IOUT,SP 00016630
80      FORMAT('1(NLSOL):',8X,A<NB>//' N=',4X,I6,T18,'K=',4X,I6,T34,'IP=',00016640
     1 3X,I6,T50,'M=',4X,I6,T66,'IALT=',1X,I6/' ISTOP=',I6,T18,'IWT=',00016650
     2 2X,I6,T34,'IDER=',I7,T50,'IPRT=',I7,T66,'NITER=',I6/' IOUT=',00016660
     3 5X,I2,T18,'SP=',3X,I6)                                       00016670
        IF(IOUT.NE.0)                                               00016680
     1WRITE(16,80)TITLE,N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,IOUT,SP 00016690
        IF(IP.GT.0) THEN                                           00016700
          WRITE(6,90) (IB(I),I=1,IP)                                00016710
90        FORMAT(/' PARAMETERS HELD FIXED: IB=',20I3)              00016720
          IF(IOUT.NE.0) WRITE(16,90) (IB(I),I=1,IP)                00016730
        ENDIF                                                       00016740
        CALL NONBLANK(FMT,NB)                                       00016750
        WRITE(6,100) FMT                                            00016760
100     FORMAT(/' FMT=',A<NB>/)                                     00016770
        IF(IOUT.NE.0) WRITE(16,100) FMT                            00016780
        IF(SP.GT.2) THEN                                           00016790
          WRITE(6,111) (BL(I),I=1,KPARMS)                          00016800
111       FORMAT(/' PARAMETER LOWER BOUNDS: BL='//(5E16.8))        00016810
          IF(IOUT.NE.0) WRITE(16,111) (BL(I),I=1,KPARMS)           00016820
        ENDIF                                                       00016830
```

```
        WRITE(6,110) (B(I),I=1,KPARMS)                           00016840
110     FORMAT(/' INITIAL PARAMETERS: B='//(5E16.8))             00016850
        IF(IOUT.NE.0) WRITE(16,110) (B(I),I=1,KPARMS)            00016860
        IF(SP.GT.2) THEN                                         00016870
          WRITE(6,112) (BH(I),I=1,KPARMS)                        00016880
112       FORMAT(/' PARAMETER HIGHER BOUNDS: BH='//(5E16.8))     00016890
          IF(IOUT.NE.0) WRITE(16,112) (BH(I),I=1,KPARMS)         00016900
        ENDIF                                                    00016910
        DO 120 I=1,KDIM                                          00016920
120     INDEX(I)=I                                               00016930
        IF(IP.EQ.0) THEN                                         00016940
          DO 130 I=1,KPARMS                                      00016950
          IF(SP.GT.2) THEN                                       00016960
            CL(I)=BL(I)                                          00016970
            CH(I)=BH(I)                                          00016980
          ENDIF                                                  00016990
130       C(I)=B(I)                                              00017000
        ELSE                                                     00017010
C                                                                00017020
C   REORDER B TO C WHEN IP>0 (AND BL,BH TO CL,CH, RESPECTIVELY)  00017030
C                                                                00017040
          IM=0                                                   00017050
          DO 150 I=1,KPARMS                                      00017060
          DO 140 J=1,IP                                          00017070
            IF(I.EQ.IB(J)) GO TO 150                             00017080
140       CONTINUE                                               00017090
          IM=IM+1                                                00017100
          C(IM)=B(I)                                             00017110
          IF(SP.GT.2) THEN                                       00017120
            CL(IM)=BL(I)                                         00017130
            CH(IM)=BH(I)                                         00017140
          ENDIF                                                  00017150
          INDEX(IM)=I                                            00017160
150       CONTINUE                                               00017170
          WRITE(6,160) (I,I=1,KPARMS)                            00017180
160       FORMAT(/' PARAMETER INDEX:',20I3)                      00017190
          IF(IOUT.NE.0) WRITE(16,160) (I,I=1,KPARMS)             00017200
          WRITE(6,170) (INDEX(I),I=1,KIP)                        00017210
170       FORMAT(' REORDERED AS...:',20I3)                       00017220
          IF(IOUT.NE.0) WRITE(16,170) (INDEX(I),I=1,KIP)         00017230
          WRITE(6,180) (C(I),I=1,KIP)                            00017240
180       FORMAT(/' REORDERED PARAMETERS:'//(5E16.8))            00017250
          IF(IOUT.NE.0) WRITE(16,180) (C(I),I=1,KIP)             00017260
        ENDIF                                                    00017270
C                                                                00017280
C   PERFORM INITIAL PARAMETER TRANSFORMS VIA SP (SCALEP)         00017290
C                                                                00017300
        IF(SP.EQ.0) GO TO 220                                    00017310
        DO 210 I=1,KIP                                           00017320
          GO TO (201,202,203,203),SP                             00017330
201       C(I)=ALOG(C(I))                                        00017340
          GO TO 210                                              00017350
202       C(I)=ASINH(C(I))                                       00017360
          GO TO 210                                              00017370
203       TEM=(C(I)-CL(I))/(CH(I)-CL(I))                         00017380
```

```
            IF(SP.EQ.3) THEN                                            00017390
              C(I)=ASIN(SQRT(TEM))                                      00017400
            ELSE                                                        00017410
              C(I)=ERFINV(2.0*TEM-1.0)                                  00017420
            ENDIF                                                       00017430
210     CONTINUE                                                       00017440
C                                                                      00017450
C   INTERFACE WITH NL2ITR USING MARQRT FCODE AND PCODE (IF IDER=0)      00017460
C                                                                      00017470
220     ENCODE(6,222,CALLED) ICALL                                     00017480
222     FORMAT(I3,' **')                                               00017490
        WRITE(6,221) CALLED                                            00017500
221     FORMAT('0** NLITR (IDER=0) OR NL2SNO (IDER=1) CALLED:',A6/)     00017510
        IF(IOUT.NE.0) WRITE(16,221) CALLED                             00017520
        IF(IDER.EQ.0) THEN                                             00017530
          CALL NLITR(NOBS,KIP,C,IV,V,CBOUND,FCODE,PCODE)               00017540
C         ***********************************************              00017550
        ELSE                                                           00017560
          CALL NL2SNO(NOBS,KIP,C,CALCR,IV,V,IDUMMY,CBOUND,FCODE)       00017570
C         *******************************************************      00017580
        ENDIF                                                          00017590
C                                                                      00017600
C   GET INVERSE PARAMETER TRANSFORMATION OF SOLUTION VECTOR C           00017610
C                                                                      00017620
        IF(SP.EQ.0) GO TO 229                                          00017630
        DO 228 I=1,KIP                                                 00017640
          GO TO (224,225,226,226),SP                                   00017650
224       C(I)=EXP(C(I))                                               00017660
          GO TO 228                                                    00017670
225       C(I)=SINH(C(I))                                              00017680
          GO TO 228                                                    00017690
226       TEM=CH(I)-CL(I)                                              00017700
          IF(SP.EQ.3) THEN                                             00017710
            C(I)=CL(I)+TEM*SIN(C(I))**2                                00017720
          ELSE                                                         00017730
            C(I)=CL(I)+0.5*TEM*(1.0+ERF(C(I)))                         00017740
          ENDIF                                                        00017750
228     CONTINUE                                                       00017760
C                                                                      00017770
C   OUTPUT SELECTED RESULTS ON FOR006 (ALL RESULTS ON FOR016 IF IOUT=1) 00017780
C                                                                      00017790
229     IF(IOUT.NE.0.AND.IPRT.NE.0) THEN                               00017800
          I=1                                                          00017810
          REWIND 16                                                    00017820
230       READ(16,232,END=240) LINE132                                 00017830
232       FORMAT(A)                                                    00017840
          IF(I.EQ.1) THEN                                              00017850
C                                                                      00017860
C   VAX FUNCTION 'LIBSINDEX' USED TO DISTINGUISH FROM ARRAY 'INDEX'     00017870
C                                                                      00017880
            IF(LIBSINDEX(LINE132,'CALLED:'//CALLED).EQ.0) GO TO 230    00017890
            I=0                                                        00017900
            GO TO 230                                                  00017910
          ENDIF                                                        00017920
          IF(LIBSINDEX(LINE132,'OBS.Y(I)').NE.0) GO TO 236             00017930
```

```
        IF(LIBSINDEX(LINE132,'COVARIANCE = SCALE').NE.0) GO TO 236    00017940
        CALL NONBLANK(LINE132,J)                                      00017950
        IF(J.LE.0) GO TO 230                                          00017960
        WRITE(6,234) LINE132                                          00017970
234     FORMAT(A<J>)                                                  00017980
        GO TO 230                                                     00017990
236     READ(16,232,END=240) LINE132                                 00018000
        GO TO 236                                                     00018010
      ENDIF                                                           00018020
240   IF(IOUT.NE.0) WRITE(16,250)                                    00018030
250   FORMAT(/3X,'I',4X,'OBS.Y(I)',6X,'CAL',11X,'RES',8X,           00018040
     1 '%RES.ERR',6X,'X(I,1)',8X,                                    00018050
     2 'X(I,2)',8X,'X(I,3)',8X,'X(I,4)',8X,'WT(I)')                  00018060
      IF(IPRT.EQ.-2) WRITE(6,250)                                    00018070
      SUMF2=0.0                                                      00018080
      IF(IDER.NE.0) IADR=IV(50)-1                                    00018090
      DO 270 I=1,NOBS                                                00018100
        IF(IDER.EQ.0) THEN                                           00018110
          F2=R(I)                                                    00018120
        ELSE                                                         00018130
          F2=V(IADR+I)                                               00018140
        ENDIF                                                        00018150
        RES=F2/SQWT(I)                                               00018160
        CAL=Y(I)-RES                                                 00018170
        IF(CAL.NE.0.0) THEN                                          00018180
          PERR=100.0*RES/ABS(CAL)                                    00018190
        ELSE                                                         00018200
          PERR=0.0                                                   00018210
        ENDIF                                                        00018220
        WT=SQWT(I)**2                                                00018230
        SUMF2=SUMF2+RES**2                                           00018240
        IF(IPRT.EQ.-2)WRITE(6,260) I,Y(I),CAL,RES,PERR,             00018250
     1 (X(I,J),J=1,4),WT                                            00018260
260     FORMAT(1X,I3,2E14.6,E11.3,6E14.6)                           00018270
        IF(IOUT.NE.0) WRITE(16,260) I,Y(I),CAL,RES,PERR,            00018280
     1 (X(I,J),J=1,4),WT                                            00018290
270   CONTINUE                                                      00018300
      IF(NOBS.EQ.KIP) THEN                                          00018310
        RMSERR=0.0                                                  00018320
      ELSE                                                          00018330
        RMSERR=SQRT(SUMF2/(NOBS-KIP))                               00018340
      ENDIF                                                         00018350
      WRITE(6,280) RMSERR                                          00018360
280   FORMAT(/' ** RMSERR=',E16.8)                                 00018370
      IF(IOUT.NE.0) WRITE(16,280) RMSERR                           00018380
      IF(IV(26).LE.0) GO TO 380                                     00018390
C                                                                   00018400
C  A COVARIANCE MATRIX WAS COMPUTED (GET ADDITIONAL STATISTICS)     00018410
C                                                                   00018420
      IADR=IV(26)-1                                                 00018430
      IF(IPRT.LT.-1) WRITE(6,290)                                   00018440
290   FORMAT(/' COVARIANCE MATRIX')                                00018450
      DO 320 I=1,KIP                                                00018460
      DO 300 J=1,I                                                  00018470
300     W(J)=V(IADR+LOC(J,I))                                      00018480
```

```
          SE(I)=SQRT(ABS(W(I)))                                      00018490
          IF(IPRT.LT.-1) WRITE(6,310) INDEX(I),(W(J),J=1,I)          00018500
310       FORMAT(1X,I2,10E12.4/(3X,10E12.4))                         00018510
320    CONTINUE                                                      00018520
C                                                                    00018530
C   GET CORRELATION COEFFICIENT MATRIX                               00018540
C                                                                    00018550
       IF(IOUT.NE.0) WRITE(16,330)                                   00018560
330    FORMAT(/' CORRELATION MATRIX')                                00018570
       IF(IPRT.LT.0) WRITE(6,330)                                    00018580
       DO 350 I=1,KIP                                                00018590
          IF(SE(I).EQ.0.0) THEN                                      00018600
            W(I)=1.0                                                 00018610
          ENDIF                                                      00018620
          DO 340 J=1,I                                               00018630
          IF(SE(J).NE.0.0) W(J)=V(IADR+LOC(J,I))/(SE(I)*SE(J))       00018640
340       CONTINUE                                                   00018650
          IF(IOUT.NE.0) WRITE(16,310) INDEX(I),(W(J),J=1,I)          00018660
          IF(IPRT.LT.0) WRITE(6,310) INDEX(I),(W(J),J=1,I)           00018670
350    CONTINUE                                                      00018680
C                                                                    00018690
C   PRINT PARAMETER STANDARD ERRORS (SE) AND RELATIVE ERRORS         00018700
C                                                                    00018710
       WRITE(6,360)                                                  00018720
360    FORMAT(/' **PARM_SOL.   STD_ERROR   REL_ERROR   % ERROR **'/) 00018730
       IF(IOUT.NE.0) WRITE(16,360)                                   00018740
       DO 370 I=1,KIP                                                00018750
          RELERR=0.0                                                 00018760
          IF(C(I).NE.0.0) RELERR=SE(I)/C(I)                          00018770
          PERR=100.*RELERR                                           00018780
          WRITE(6,310) INDEX(I),C(I),SE(I),RELERR,PERR               00018790
          IF(IOUT.NE.0) WRITE(16,310) INDEX(I),C(I),SE(I),RELERR,PERR 00018800
370    CONTINUE                                                      00018810
C                                                                    00018820
C   PUT SOLUTION C AND BFIX TOGETHER (IF IP>0)                       00018830
C                                                                    00018840
380    DO 390 I=1,KIP                                                00018850
390    W(I)=C(I)                                                     00018860
       IF(IP.EQ.0) GO TO 420                                         00018870
       IM=0                                                          00018880
       DO 410 I=1,KPARMS                                             00018890
          W(I)=BFIX(I)                                               00018900
          DO 400 J=1,IP                                              00018910
          IF(I.EQ.IB(J)) GO TO 410                                   00018920
400       CONTINUE                                                   00018930
          IM=IM+1                                                    00018940
          W(I)=C(IM)                                                 00018950
410    CONTINUE                                                      00018960
420    CALL SUBEND(Y,X,W,K,N,TITLE,IOUT)                             00018970
C      *********************************                             00018980
       IF(ISTOP.NE.1) THEN                                           00018990
          READ(5,10,ERR=9000,END=9010) TITLE                        00019000
          IF(IALT.NE.5) REWIND IALT                                  00019010
          ICALL=ICALL+1                                              00019020
          GO TO 22                                                   00019030
```

```
      ENDIF                                                           00019040
C                                                                     00019050
C** RETURN FROM NLSOL                                                 00019060
C                                                                     00019070
      RETURN                                                          00019080
9000  CALL ERRMSG('ERR=9000 READING FOR005',0,6,16)                  00019090
9010  CALL ERRMSG('PREMATURE E.O.F (END=9010) READING FOR005',0,6,16) 00019100
9020  CALL ERRMSG('END #9020 READING FOR005 IN (NAMELIST)',0,6,16)   00019110
9030  CALL ERRMSG('END=9030 READING FILE IALT',0,6,16)               00019120
9040  CALL ERRMSG('PREMATURE E.O.F (END=9040) READING FILE IALT',     00019130
     1 0,6,16)                                                        00019140
C                                                                     00019150
C** END OF SUBROUTINE NLSOL                                           00019160
C                                                                     00019170
      END                                                             00019180
      SUBROUTINE NLITR(N,KIP,C,IV,V,CBOUND,FCODE,PCODE)               00019190
C                                                                     00019200
C**CALCULATES BOTH THE RESIDUAL VECTOR R(N) & ANALYTICAL JACOBIAN     00019210
C  JAC(N,KIP) BY 'REVERSE COMMUNICATION VIA INTERNAL CALL NL2ITR'     00019220
C  (SEE REF1, P. 38).                                                 00019230
C                                                                     00019240
C     N = NO. OBSERVATIONS <=500 (SEE NDIM BELOW)                     00019250
C     KIP = NO. ADJUSTABLE PARAMETERS =K-IIP WHERE                    00019260
C        K=TOTAL PARAMETERS, IIP=NO. PARAMETERS HELD FIXED            00019270
C        IN IIB(IIP) VIA COMMON/FIXDAT/                               00019280
C     C() = I/O PARAMETER VECTOR (SUPPLIED BY NL2ITR)                 00019290
C        WHICH ARE THE UNCONSTRAINED PARAMETERS IN NL2ITR.            00019300
C     IV() = SAME CONTROL INFORMATION SET BY NLSOL (OR NL2ITR).       00019310
C     V() = SAME CONTROL INFORMATION SET BY NLSOL (OR NL2ITR).        00019320
C     CBOUND = INPUT ARRAY OF LOW AND HIGH BOUNDS USED ONLY WHEN SP>2. 00019330
C     FCODE = EXTERNAL FUNCTION NAME (SAME AS USED IN 'MARQRT' OR     00019340
C        'IMSLMQ' TO COMPUTE THE NONLINEAR OBJECTIVE FUNCTION).       00019350
C     PCODE = EXTERNAL ANALYTIC DERIVATIVE NAME (SAME AS USED IN      00019360
C        'MARQRT' WHEN IDER=0) CORRESPONDING TO EACH FCODE CALL.      00019370
C                                                                     00019380
C**SEE REF1 (P.38) FOR MORE DETAILS ON CALLING NL2ITR.               00019390
C                                                                     00019400
C**OTHER DATA IN COMMON/FIXDAT/ MUST BE PRESET.                       00019410
C                                                                     00019420
C                                                                     00019430
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00019440
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF       00019450
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                      00019460
C     PARAMETER (NDIM=500,MDIM=5,KDIM=20)                             00019470
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS. 00019480
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:          00019490
C     PARAMETER (K1DIM=KDIM-1)                                        00019500
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00019510
C                                                                     00019520
      INTEGER SP                                                      00019530
      DIMENSION C(1),IV(1),V(1),CBOUND(1),PRNT(5),SQWT(NDIM),         00019540
     1 BIP(KDIM),D(KDIM),R(NDIM),PART(KDIM),W(KDIM)                   00019550
      REAL*4 JAC(NDIM,KDIM)                                           00019560
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,   00019570
     1 IDER,KPARMS,SP                                                 00019580
```

```
      COMMON/BOUNDS/BL(KDIM),BH(KDIM)                              00019590
      COMMON/REVCOM/R                                             00019600
      EQUIVALENCE (SQWT(1),X(1,MDIM))                             00019610
      DATA NN/NDIM/                                               00019620
C                                                                 00019630
C  GET INVERSE PARAMETER TRANSFORMATION (C TO BIP)                00019640
C                                                                 00019650
10    CALL INTRAN(KIP,C,CBOUND,BIP)                               00019660
C                                                                 00019670
C  DETERMINE FROM IV(1) HOW TO CALL NL2ITR                        00019680
      IV1=IV(1)                                                   00019690
         DO 120 I=1,N                                             00019700
           CALL FCODE(Y,X,BIP,PRNT,F,I,IDER)                      00019710
C          *****************************                          00019720
           IF(IV1.NE.2) R(I)=SQWT(I)*(Y(I)-F)                     00019730
           IF(IV1.EQ.1) GO TO 120                                 00019740
           CALL PCODE(PART,X,BIP,PRNT,F,I,IIP,IIB)                00019750
C          ***************************************                00019760
C                                                                 00019770
C  SCALE PART(J) VIA SP AND THE DERIVATIVE CHAIN-RULE.            00019780
C                                                                 00019790
           IF(SP.EQ.0) GO TO 80                                   00019800
           IF(SP.EQ.1) THEN                                       00019810
             DO 11 K=1,KPARMS                                     00019820
11           PART(K)=BIP(K)*PART(K)                               00019830
           ELSE IF(SP.EQ.2) THEN                                  00019840
             DO 12 K=1,KPARMS                                     00019850
             IF(PART(K).EQ.0.0) GO TO 12                          00019860
             TEM=BIP(K)+SQRT(BIP(K)**2+1.0)                       00019870
             PART(K)=0.5*(TEM+1.0/TEM)*PART(K)                    00019880
12           CONTINUE                                             00019890
           ELSE IF (SP.EQ.3) THEN                                 00019900
             DO 13 K=1,KPARMS                                     00019910
             IF(PART(K).EQ.0.0) GO TO 13                          00019920
             PART(K)=2.*PART(K)*SQRT((BIP(K)-BL(K))*              00019930
     1         (BH(K)-BIP(K)))                                    00019940
13           CONTINUE                                             00019950
           ELSE IF(SP.EQ.4) THEN                                  00019960
             DO 14 K=1,KPARMS                                     00019970
             IF(PART(K).EQ.0.0) GO TO 14                          00019980
             TEM=BH(K)-BL(K)                                      00019990
             PART(K)=0.56418958*PART(K)*TEM*EXP(-(ERFINV(2.*(BIP(K)- 00020000
     1         BL(K))/TEM-1.))**2)                                00020010
14           CONTINUE                                             00020020
           ENDIF                                                 00020030
80         IF(IIP.EQ.0) THEN                                     00020040
             DO 90 J=1,KIP                                       00020050
90           JAC(I,J)=-SQWT(I)*PART(J)                           00020060
           ELSE                                                  00020070
             IM=0                                                00020080
             DO 110 K=1,KPARMS                                   00020090
             DO 100 J=1,IIP                                      00020100
               IF(K.EQ.IIB(J)) GO TO 110                         00020110
100          CONTINUE                                            00020120
             IM=IM+1                                             00020130
```

```
              JAC(I,IM)=-SQwT(I)*PART(K)                        00020140
110          CONTINUE                                          00020150
         ENDIF                                                 00020160
120     CONTINUE                                               00020170
C                                                              00020180
C                                                              00020190
      CALL NL2ITR(D,IV,JAC,N,NN,KIP,R,V,C)                     00020200
C     *****************************************                00020210
      IF(IV(1).EQ.1.OR.IV(1).EQ.2) GO TO 10                    00020220
      RETURN                                                   00020230
      END                                                      00020240
      SUBROUTINE INTRAN(KIP,C,CBOUND,BIP)                      00020250
C                                                              00020260
C**INVERSE PARAMETER TRANSFORMATION USED IN 'NLSOL','NLITR'.   00020270
C                                                              00020280
C  CALCULATES CONSTRAINED PARAMETERS FOR FCODE OR PCODE BACK FROM THE   00020290
C  UNCONSTRAINED PARAMETERS IN 'NL2ITR' OR 'NL2SNO'            00020300
C                                                              00020310
C     KIP = NO. ADJUSTABLE PARAMETERS = K-IIP (IIP IN COMMON/FIXDAT)    00020320
C     C() = INPUT UNCONSTRAINED VECTOR (DIM. KIP)             00020330
C     CBOUND = INPUT CONSTRAINED BOUNDS, IF ANY.              00020340
C     BIP() = OUTPUT CONSTRAINED VECTOR (DIM. KPARMS--IN COMMON).       00020350
C                                                              00020360
CSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS00020370
CSS CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF 00020380
CSS INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:              00020390
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                     00020400
CSS WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.  00020410
CSS DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:  00020420
      PARAMETER (K1DIM=KDIM-1)                                00020430
CSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS00020440
C                                                              00020450
      INTEGER SP                                               00020460
      DIMENSION C(1),CBOUND(1),BIP(1),CTEM(KDIM)               00020470
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,      00020480
     1 IDER,KPARMS,SP                                          00020490
      IF(SP.EQ.0) THEN                                         00020500
         DO 10 I=1,KIP                                         00020510
10       CTEM(I)=C(I)                                          00020520
      ELSE                                                     00020530
         DO 50 I=1,KIP                                         00020540
            GO TO (20,30,40,40),SP                             00020550
20          CTEM(I)=EXP(C(I))                                  00020560
            GO TO 50                                           00020570
30          CTEM(I)=SINH(C(I))                                 00020580
            GO TO 50                                           00020590
40          DIF=CBOUND(KDIM+I)-CBOUND(I)                       00020600
            IF(SP.EQ.3) THEN                                   00020610
               CTEM(I)=CBOUND(I)+DIF*SIN(C(I))**2              00020620
            ELSE                                               00020630
               CTEM(I)=CBOUND(I)+0.5*DIF*(1.0+ERF(C(I)))       00020640
            ENDIF                                              00020650
50       CONTINUE                                              00020660
      ENDIF                                                    00020670
      IF(IIP.EQ.0) THEN                                        00020680
```

```
           DO 60 I=1,KIP                                          00020690
60         BIP(I)=CTEM(I)                                         00020700
        ELSE                                                      00020710
          IM=0                                                    00020720
          DO 80 I=1,KPARMS                                        00020730
            BIP(I)=BFIX(I)                                        00020740
          DO 70 J=1,IIP                                           00020750
            IF(I.EQ.IIB(J)) GO TO 80                              00020760
70         CONTINUE                                               00020770
          IM=IM+1                                                 00020780
          BIP(I)=CTEM(IM)                                         00020790
80         CONTINUE                                               00020800
        ENDIF                                                     00020810
        RETURN                                                    00020820
        END                                                       00020830
        SUBROUTINE CALCR(N,KIP,C,NF,R,LASTNF,CBOUND,FCODE)        00020840
C                                                                 00020850
C**CALCULATES RESIDUAL VECTOR R(N) FOR 'NL2SNO' WHEN IDER=1.      00020860
C                                                                 00020870
C     N = NO. OBSERVATIONS <=500 (SEE NDIM BELOW)                 00020880
C     KIP = NO. ADJUSTABLE PARAMETERS =K-IIP WHERE                00020890
C        K=TOTAL PARAMETERS, IIP=NO. PARAMETERS HELD FIXED        00020900
C          IN IIB(IIP) VIA COMMON/FIXDAT/                         00020910
C     C() = INPUT PARAMETER VECTOR (SUPPLIED BY NL2SNO)           00020920
C        WHICH ARE THE UNCONSTRAINED PARAMETERS IN NL2SNO.        00020930
C     NF = INVOCATION COUNT (INPUT)FOR USE BY NL2SNO OR NL2SOL.   00020940
C     R() = OUTPUT WEIGHTED RESIDUAL VECTOR (DIM. N)              00020950
C     LASTNF = LAST NF (ON EXIT FOR POSSIBLE USE IN CALCJ OR NL2SOL).  00020960
C     CBOUND = INPUT ARRAY OF LOW AND HIGH BOUNDS USED ONLY WHEN SP>2. 00020970
C     FCODE = EXTERNAL FUNCTION NAME (SAME AS USED IN 'MARQRT' OR 00020980
C        'IMSLMQ' TO COMPUTE THE NONLINEAR OBJECTIVE FUNCTION).   00020990
C                                                                 00021000
C**OTHER DATA IN COMMON/FIXDAT/ MUST BE PRESET.                   00021010
C                                                                 00021020
C                                                                 00021030
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00021040
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF   00021050
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                  00021060
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                         00021070
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS. 00021080
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:      00021090
      PARAMETER (K1DIM=KDIM-1)                                    00021100
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00021110
C                                                                 00021120
      INTEGER SP                                                  00021130
      DIMENSION C(1),R(1),CBOUND(1),PRNT(5),SQWT(NDIM),BIP(KDIM)  00021140
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,  00021150
     1 IDER,KPARMS,SP                                             00021160
      EQUIVALENCE (SQWT(1),X(1,MDIM))                             00021170
C                                                                 00021180
C  GET INVERSE PARAMETER TRANSFORMATION (C TO BIP)                00021190
C                                                                 00021200
      CALL INTRAN(KIP,C,CBOUND,BIP)                               00021210
C                                                                 00021220
C  COMPUTE RESIDUAL VECTOR R(N) USING BIP IN FCODE                00021230
```

```
C                                                                        00021240
      DO 10 I=1,N                                                        00021250
        CALL FCODE(Y,X,BIP,PRNT,F,I,IDER)                               00021260
C       *********************************                              00021270
        R(I)=SQWT(I)*(Y(I)-F)                                           00021280
10    CONTINUE                                                          00021290
      LASTNF=NF                                                         00021300
      RETURN                                                            00021310
      END                                                               00021320
      SUBROUTINE NONBLANK(C,NB)                                         00021330
C--DETERMINE NON-BLANK CHAR LENGTH (=NB ON EXIT) OF C*(*)              00021340
C  NOTE THAT NB WILL BE IN [0,LEN(C)].                                 00021350
C                                                                       00021360
      CHARACTER*(*) C                                                   00021370
      L=LEN(C)                                                          00021380
      DO 10 I=L,1,-1                                                    00021390
        NB=I                                                           00021400
        IF(C(I:I).NE.' ') RETURN                                       00021410
10    CONTINUE                                                          00021420
      NB=0                                                             00021430
      RETURN                                                            00021440
      END                                                               00021450
      SUBROUTINE PROCINFO(ABS_VALUES,INCR_VALUES)                      00021460
C                                                                       00021470
C** SUBROUTINE TO OBTAIN ABSOLUTE AND INCREMENTAL VALUES OF PROCESS    00021480
C   PARAMETERS: CPU TIME, BUFFERED I/O COUNT, DIRECT I/O COUNT, AND    00021490
C   PAGE FAULTS.                                                       00021500
C                                                                       00021510
      IMPLICIT INTEGER*2(W),INTEGER*4(L)                               00021520
      PARAMETER (JPIS_CPUTIM = '00000407'X,                            00021530
     1 JPIS_BUFIO = '0000040C'X,JPIS_DIRIO = '0000040B'X,              00021540
     2 JPIS_PAGEFLTS= '0000040A'X)                                     00021550
      INTEGER*4 ABS_VALUES(4),INCR_VALUES(4),LCL_VALUES(4)             00021560
      COMMON/ITEMLIST/                                                  00021570
     1 W_LEN1,W_CODE1,L_ADDR1,L_LENADDR1,                              00021580
     2 W_LEN2,W_CODE2,L_ADDR2,L_LENADDR2,                              00021590
     3 W_LEN3,W_CODE3,L_ADDR3,L_LENADDR3,                              00021600
     4 W_LEN4,W_CODE4,L_ADDR4,L_LENADDR4,                              00021610
     5 W_LEN5,W_CODE5                                                  00021620
      DATA W_LEN1,W_LEN2,W_LEN3,W_LEN4,W_LEN5/5*4/                     00021630
      DATA W_CODE1/JPIS_CPUTIM/,                                       00021640
     1 W_CODE2/JPIS_BUFIO/,                                            00021650
     2 W_CODE3/JPIS_DIRIO/,                                            00021660
     3 W_CODE4/JPIS_PAGEFLTS/,                                         00021670
     4 W_CODE5/0/                                                      00021680
      DATA L_LENADDR1,L_LENADDR2,L_LENADDR3,L_LENADDR4/4*0/            00021690
      L_ADDR1=%LOC(LCL_VALUES(1))                                     00021700
      L_ADDR2=%LOC(LCL_VALUES(2))                                     00021710
      L_ADDR3=%LOC(LCL_VALUES(3))                                     00021720
      L_ADDR4=%LOC(LCL_VALUES(4))                                     00021730
C** PERFORM THE SYSTEM SERVICE CALL                                   00021740
      CALL SYS$GETJPI(,,,W_LEN1,,,)                                   00021750
C** ASSIGN THE NEW VALUES TO THE ARGUMENTS                            00021760
      DO I=1,4                                                        00021770
        INCR_VALUES(I)=LCL_VALUES(I)-ABS_VALUES(I)                    00021780
```

```
        ABS_VALUES(I)=LCL_VALUES(I)                                 00021790
      END DO                                                        00021800
      RETURN                                                        00021810
      END                                                           00021820
      REAL FUNCTION RFLAGS(N,FUN,TOL,TO,TM,T,NEW)                   00021830
C--FOURIER TRANSFORM LAG CONVOLUTION & SPLINE INTERPOLATION         00021840
C   GIVES FOURIER COSINE OR SINE TRANSFORMS VIA RLAGFO,RLAGF1       00021850
C   REF: ANDERSON,1975,NTIS REPT. PB-242-800,P.76-87.              00021860
C                                                                   00021870
C      N = 0 FOR COSINE TRANSFORM (VIA RLAGFO)                      00021880
C      N = 1 FOR SINE TRANSFORM (VIA RLAGF1)                        00021890
C    FUN = EXTERNAL REAL KERNEL FUNCTION.                           00021900
C    TOL = TOLERANCE REQUESTED FOR RLAGFO OR RLAGF1                 00021910
C     TO = TMIN TO USE (E.G., LET TO=.5*TMIN, TMIN=TRUE)           00021920
C     TM = TMAX TO USE (TM>TO)                                      00021930
C      T = TRANSFORM PARAMETER (TO<=T<=TM) FOR THIS CALL (NEW=1 OR 0) 00021940
C    NEW = 1 REQUIRED FOR 1ST CALL OR TO RESET SPLINE COEFFICIENTS. 00021950
C    NEW = 0 FOR ALL CALLS AFTER 1ST--USES SPLINE INTERPOLATION ONLY. 00021960
C                                                                   00021970
      REAL ARG(200),Y(200),AR(200),BR(200),CR(200),                00021980
     & D(2),W1(200),W2(200)                                         00021990
      EXTERNAL FUN                                                  00022000
      DATA D/2*0.0/                                                 00022010
      IF(NEW.EQ.0) GO TO 3                                          00022020
      NT=AINT(5.*ALOG(TM/TO))+5                                     00022030
      IF(NT.GT.200)CALL ERRMSG('IN RFLAGS: NT>200      ',4,6,16)    00022040
      NT1=NT+1                                                      00022050
      XO=ALOG(TO)+.2*NT                                             00022060
      NU=1                                                          00022070
      DO 1 J=1,NT                                                   00022080
      I=NT1-J                                                       00022090
      X=XO-.2*J                                                     00022100
      EX=EXP(X)                                                     00022110
      ARG(I)=EX                                                     00022120
      IF(N.EQ.0) Y(I)=RLAGFO(X,FUN,TOL,L,NU)/EX                     00022130
      IF(N.NE.0) Y(I)=RLAGF1(X,FUN,TOL,L,NU)/EX                     00022140
1     NU=0                                                          00022150
      CALL SPLIN1(NT,0.0,ARG,Y,AR,BR,CR,0,D,W1,W2)                  00022160
2     IF(NT.LT.0) CALL ERRMSG('IN RFLAGS: NT<0 AFTER SPLIN1   ',6,6,16) 00022170
3     IF(T.LT.TO) CALL ERRMSG('IN RFLAGS: T<TO',3,6,16)            00022180
      IF(T.GT.TM) CALL ERRMSG('IN RFLAGS: T>TM',3,6,16)            00022190
      CALL SPOINT(NT,ARG,Y,AR,BR,CR,T,X)                           00022200
      RFLAGS=X                                                      00022210
      RETURN                                                        00022220
      END                                                           00022230
      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                     00022240
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION.          00022250
C                                                                   00022260
C          BY   W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO 00022270
C                                                                   00022280
C   PARMS--- M= NUMBER OF DATA POINTS .GT. 2                        00022290
C            H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE), 00022300
C               UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE)      00022310
C            X= INDEP.VAR WHEN H=0. (DIM .GE. M).                   00022320
C            Y= DEPENDENT VARIABLE  (DIM .GE. M).                   00022330
```

```
C             A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)                          00022340
C                   RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C.    00022350
C                   ALSO USED AS WORK ARRAYS DURING EXECUTION.              00022360
C             IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE       00022370
C                IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR            00022380
C                IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS.               00022390
C             D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY.      00022400
C             P,S= WORK ARRAYS (EACH DIM=M).                                00022410
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.                  00022420
C   THE RESULTING CUBIC SPLINE IS OF THE FORM:                             00022430
C      Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3              00022440
C        FOR I=1,2,...,M-1                                                  00022450
C                                                                          00022460
C                                                                          00022470
      REAL*4  X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL                   00022480
      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999                00022490
      N=M-1                                                                 00022500
      IF(IT.EQ.0) GO TO 20                                                  00022510
C--1ST DERIVATIVE BOUNDARIES GIVEN                                          00022520
      NE=N-1                                                                00022530
      IF(H) 999,11,1                                                        00022540
C--EQUAL SPACING H .GT. 0. AND IT=1                                         00022550
    1 HH=3.0/H                                                              00022560
      DO 2 I=1,NE                                                           00022570
      B(I)=4.0                                                              00022580
      C(I)=1.0                                                              00022590
      A(I)=1.0                                                              00022600
    2 P(I)=HH*(Y(I+2)-Y(I))                                                 00022610
      P(1)=P(1)-D(1)                                                        00022620
      P(NE)=P(NE)-D(2)                                                      00022630
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE                           00022640
    3 C(1)=C(1)/B(1)                                                        00022650
      P(1)=P(1)/B(1)                                                        00022660
      DO 4 I=2,NE                                                           00022670
      MUL=1.0/(B(I)-A(I)*C(I-1))                                            00022680
      C(I)=MUL*C(I)                                                         00022690
    4 P(I)=MUL*(P(I)-A(I)*P(I-1))                                           00022700
C--OBTAIN SPLINE COEFFICIENTS                                               00022710
      A(NE+IT)=P(NE)                                                        00022720
      I=NE-1                                                                00022730
    5 A(I+IT)=P(I)-C(I)*A(I+IT+1)                                           00022740
      I=I-1                                                                 00022750
       IF(I.GE.1) GO TO 5                                                   00022760
      IF(IT.EQ.0) GO TO 6                                                   00022770
      A(1)=D(1)                                                             00022780
      A(M)=D(2)                                                             00022790
    6 IF(H.EQ.0.) GO TO 14                                                  00022800
      HH=1.0/H                                                              00022810
      DO 7 I=1,N                                                            00022820
      MUL=HH*(Y(I+1)-Y(I))                                                  00022830
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))                                   00022840
    7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))                                     00022850
      RETURN                                                                00022860
C--UNEQUAL SPACING H=0.. AND IT=1                                           00022870
   11 DO 12 I=1,N                                                           00022880
```

```
   12 S(I+1)=X(I+1)-X(I)                                               00022890
      DO 13 I=1,NE                                                     00022900
      B(I)=2.0*(S(I+1)+S(I+2))                                         00022910
      C(I)=S(I+1)                                                      00022920
      A(I)=S(I+2)                                                      00022930
   13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/    00022940
     $ (S(I+1)*S(I+2))                                                 00022950
      P(1)=P(1)-S(3)*D(1)                                              00022960
      P(NE)=P(NE)-S(N)*D(2)                                            00022970
      GO TO 3                                                          00022980
   14 DO 15 I=1,N                                                      00022990
      HH=1.0/S(I+1)                                                    00023000
      MUL=(Y(I+1)-Y(I))*HH**2                                          00023010
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH                                00023020
   15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2                             00023030
      RETURN                                                           00023040
C--2ND DERIVATIVE BOUNDARIES GIVEN                                     00023050
   20 NE=N+1                                                           00023060
      IF(H) 999,31,21                                                  00023070
C--EQUAL SPACING H .GT. 0 AND IT=0                                     00023080
   21 HH=3.0/H                                                         00023090
      DO 22 I=2,N                                                      00023100
      B(I)=4.0                                                         00023110
      C(I)=1.0                                                         00023120
      A(I)=1.0                                                         00023130
   22 P(I)=HH*(Y(I+1)-Y(I-1))                                          00023140
      B(1)=2.0                                                         00023150
      B(NE)=2.0                                                        00023160
      C(1)=1.0                                                         00023170
      C(NE)=1.0                                                        00023180
      A(NE)=1.0                                                        00023190
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)                                   00023200
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)                                  00023210
      GO TO 3                                                          00023220
C--UNEQUAL SPACING H=0 AND IT=0                                        00023230
   31 DO 32 I=1,N                                                      00023240
   32 S(I+1)=X(I+1)-X(I)                                               00023250
      N1=N-1                                                           00023260
      DO 33 I=1,N1                                                     00023270
      B(I+1)=2.0*(S(I+1)+S(I+2))                                       00023280
      C(I+1)=S(I+1)                                                    00023290
      A(I+1)=S(I+2)                                                    00023300
   33 P(I+1)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/  00023310
     *      (S(I+1)*S(I+2))                                            00023320
      B(1)=2.0                                                         00023330
      B(NE)=2.0                                                        00023340
      C(1)=1.0                                                         00023350
      C(NE)=1.0                                                        00023360
      A(NE)=1.0                                                        00023370
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)                          00023380
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)                         00023390
      GO TO 3                                                          00023400
  999 M=-IABS(M)                                                       00023410
      RETURN                                                           00023420
      END                                                              00023430
```

```
      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                          00023440
C--GIVEN CUBIC SPLINE COEFF'S A,B,C,AND M OBS.DATA ARRAYS X,Y       00023450
C  SPOINT EVALUATES THE PIECEWISE CUBIC SPLINE ORDINATE YY AT THE   00023460
C  ABSCISSA XX, WHERE XX IS IN THE CLOSED INTERVAL (X(1),X(M)).     00023470
C  NOTE: IF COMPUTING OVER EQUAL INTERVALS, USE THE SUBR 'CUBIC'    00023480
C  WHICH REQUIRES ONLY ONE CALL.                                    00023490
C                                                                   00023500
      DIMENSION X(1),Y(1),A(1),B(1),C(1)                            00023510
      IF(XX.LT.X(1).OR.XX.GT.X(M)) GO TO 9                          00023520
      M1=M-1                                                        00023530
      DO 1 I=1,M1                                                   00023540
      J=I                                                           00023550
      IF(XX.LE.X(I+1)) GO TO 2                                      00023560
    1 CONTINUE                                                      00023570
    9 WRITE(6,60) XX,X(1),X(M)                                      00023580
   60 FORMAT('0ERROR IN SPOINT CALL--XX=',E16.8,' NOT IN CLOSED INTERVAL00023590
     * (',E16.8,',',E16.8,')')                                      00023600
      RETURN                                                        00023610
    2 Z=XX-X(J)                                                     00023620
      YY=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z                              00023630
      RETURN                                                        00023640
      END                                                           00023650
      SUBROUTINE WARN(MSG,ISKIP,IUNIT1,IUNIT2,*)                    00023660
C                                                                   00023670
C  GENERAL WARNING MESSAGE OUTPUT AND RETURN 1 ON VAX-11/780        00023680
C                                                                   00023690
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                              00023700
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2     00023710
C          > 0 FOR ONE BLANK LINE BEFORE.                           00023720
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1). 00023730
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2). 00023740
C                                                                   00023750
C  MESSAGES ARE WRITTEN IN THE FORM:                                00023760
C                                                                   00023770
C  (WARN): _MSG_HERE_                                               00023780
C                                                                   00023790
      CHARACTER*(*) MSG                                             00023800
      I=LEN(MSG)                                                    00023810
      DO 1 J=1,2                                                    00023820
        IF(J.EQ.1) THEN                                            00023830
       JUNIT=IUNIT1                                                 00023840
          ELSE                                                     00023850
       JUNIT=IUNIT2                                                 00023860
         ENDIF                                                      00023870
        IF(JUNIT.GT.0) THEN                                        00023880
          IF(ISKIP.EQ.0) THEN                                      00023890
            WRITE(JUNIT,2) MSG                                      00023900
            ELSE                                                   00023910
       WRITE(JUNIT,3) MSG                                           00023920
            ENDIF                                                  00023930
        ENDIF                                                      00023940
    1   CONTINUE                                                   00023950
        RETURN 1                                                   00023960
    2   FORMAT(1X,'(WARN): ',A<I>)                                 00023970
    3   FORMAT(/1X,'(WARN): ',A<I>)                                00023980
```

```
      END                                                        00023990
      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                00024000
C (VAX-11/780 VERSION FORTRAN-77 (X3.9-1978); SEE NOTE(2) BELOW.)  00024010
C================================================================00024020
C  COMPLEX HANKEL TRANSFORMS OF ORDER 0 OR 1 FOR RELATED (SAVED) KERNELS00024030
C  AND FIXED TRANSFORM ARGUMENT B.GT.0.                          00024040
C                                                                00024050
C--REF: ANDERSON, W.L., 1979, GEOPHYSICS, VOL. 44, NO. 7, P. 1287-1305. 00024060
C                                                                00024070
C--SUBPROGRAM ZHANKS EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF  00024080
C  FUN(G)*JN(G*B)*DG, DEFINED AS THE COMPLEX HANKEL TRANSFORM OF   00024090
C  ORDER N (=0 OR 1) AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY  00024100
C  ADAPTIVE DIGITAL FILTERING OF THE COMPLEX KERNEL FUNCTION FUN,  00024110
C  USING DIRECT AND/OR PREVIOUSLY SAVED KERNEL FUNCTION VALUES.   00024120
C                                                                00024130
C--PARAMETERS (ALL INPUT, EXCEPT NF)                             00024140
C                                                                00024150
C     N     = ORDER (=0 OR 1) OF THE HANKEL TRANSFORM TO BE EVALUATED.  00024160
C     B     = REAL TRANSFORM ARGUMENT B.GT.0.0 OF THE HANKEL TRANSFORM. 00024170
C             IF NEW=0, B IS ASSUMED EQUAL TO THE LAST B USED WHEN NEW=100024180
C             (SEE PARAMETER NEW AND SUBPROGRAM USAGE BELOW).    00024190
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)  00024200
C             OF A REAL ARGUMENT G.GT.0. THIS REFERENCE MUST BE SUPPLIED00024210
C             EVEN WHEN NEW=0, SINCE THE ADAPTIVE CONVOLUTION     00024220
C             MAY NEED SOME DIRECT FUNCTION CALLS (E.G. IF TOL REDUCED).00024230
C             IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON00024240
C             IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  BOTH  00024250
C             REAL AND IMAGINARY PARTS OF THE COMPLEX FUNCTION FUN(G)  00024260
C             MUST BE CONTINUOUS BOUNDED FUNCTIONS FOR G.GT.0.0. FOR A  00024270
C             REAL FUNCTION F1(G), FUN=CMPLX(F1(G),0.0) MAY BE USED.  00024280
C             TWO INDEPENDENT REAL-FUNCTIONS F1(G),F2(G) MAY BE    00024290
C             INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)). 00024300
C     TOL   = REQUESTED REAL TRUNCATION TOLERANCE ACCEPTED AT THE FILTER00024310
C             TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS  00024320
C             DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS  00024330
C             THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00024340
C             TOL.LE.0.00001 WOULD GIVE ABOUT .01 PER CENT ACCURACY  00024350
C             FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR  00024360
C             VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED.  00024370
C             IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER 00024380
C             ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE 00024390
C             USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL  00024400
C             EVALUATION -- SEE NOTE (1) BELOW).                  00024410
C             FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0 MAY BE USED.  00024420
C     NF    = TOTAL NUMBER OF DIRECT FUN CALLS USED DURING CONVOLUTION  00024430
C             FOR ANY VALUE OF NEW (NF IS AN OUTPUT PARAMETER).   00024440
C             NF IS IN THE RANGE 21.LE.NF.LE.283 WHEN NEW=1.  USUALLY, 00024450
C             NF IS MUCH LESS THAN 283 (OR 0) WHEN NEW=0.        00024460
C     NEW   =1 IS REQUIRED FOR THE VERY FIRST CALL TO ZHANKS, OR IF  00024470
C             FORCING DIRECT FUNCTION FUN(G) CALLS, E.G., IF USING  00024480
C             ZHANKS FOR UNRELATED KERNELS.                      00024490
C             NEW=1 INITIALIZES COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE 00024500
C             FOR NSAVE COMPLEX KERNEL VALUES IN FSAVE AND CORRESPONDING00024510
C             REAL ARGUMENTS IN GSAVE FOR THE GIVEN PARAMETER B.  00024520
C     NEW   =0 TO USE RELATED KERNELS (MODIFIED BY USER) CURRENTLY STORED00024530
```

```
C                 IN COMMON/SAVE/. FUN IS CALLED ONLY IF REQUIRED       00024540
C                 DURING THE CONVOLUTION.  ADDITIONAL FUNCTION VALUES WHEN 00024550
C                 NEEDED ARE AUTOMATICALLY ADDED TO THE COMMON/SAVE/ BLOCK. 00024560
C                                                                      00024570
C        ******* NOTE THAT IT IS THE USERS RESPONSIBILITY TO MODIFY THE  00024580
C                 COMMON FSAVE() VALUES FOR NEW=0 CALLS, EXTERNALLY IN  00024590
C                 THE USERS CALLING PROGRAM (SEE SUBPROGRAM USAGE BELOW). 00024600
C                                                                      00024610
C=================================================================00024620
C--SUBPROGRAM USAGE-- ZHANKS IS CALLED AS FOLLOWS                       00024630
C        ...                                                           00024640
C        COMPLEX Z1,Z2,ZHANKS,FSAVE                                    00024650
C        COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                       00024660
C        EXTERNAL ZF1,ZF2                                              00024670
C        ...                                                           00024680
C        Z1=ZHANKS(N1,B,ZF1,TOL,NF1,1)                                 00024690
C        DO 1 I=1,NSAVE                                                00024700
C   C--MODIFY FSAVE IN COMMON/SAVE/ TO OBTAIN RELATED ZF2 FROM ZF1.      00024710
C   C--E.G. FSAVE(I)=GSAVE(I)*FSAVE(I) -- FOR RELATION ZF2(G)=G*ZF1(G)  00024720
C     1 CONTINUE                                                       00024730
C        Z2=ZHANKS(N2,B,ZF2,TOL,NF2,0)                                 00024740
C        ...                                                           00024750
C        END                                                          00024760
C        COMPLEX FUNCTION ZF1(G)                                       00024770
C        ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF1(G), G.GT.0.  00024780
C        END                                                          00024790
C        COMPLEX FUNCTION ZF2(G)                                       00024800
C        ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF2(G), G.GT.0.  00024810
C        END                                                          00024820
C=================================================================00024830
C--NOTES                                                               00024840
C        (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.     00024850
C              THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS 00024860
C              EXP-UNDERFLOW TO 0.0.                                   00024870
C        (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1966) IS USED, EXCEPT00024880
C              DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.00024890
C              TO CONVERT ZHANKS TO THE NEW AMERICAN STANDARD FORTRAN    00024900
C              (X3.9-1978), ADD THE FOLLOWING DECLARATION TO THIS ROUTINE00024910
C              SAVE Y1,ISAVE                                          00024920
C        (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT   00024930
C              IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),   00024940
C              BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.          00024950
C        (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME),  00024960
C              HENCE UNPREDICTABLE RESULTS COULD OCCUR IF ZHANKS         00024970
C              IS CALLED INCORRECTLY (OR IF FUN OR COMMON IS IN ERROR).  00024980
C=================================================================00024990
C                                                                      00025000
      SAVE Y1,ISAVE                                                    00025010
      COMPLEX FUN,C,CMAX,FSAVE                                         00025020
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                          00025030
      DOUBLE PRECISION E,ER,Y1,Y                                       00025040
      DIMENSION T(2),TMAX(2)                                           00025050
      DIMENSION WT0(283),WA0(76),WB0(76),WC0(76),WD0(55),              00025060
     * WT1(283),WA1(76),WB1(76),WC1(76),WD1(55)                        00025070
      EQUIVALENCE (WT0(1),WA0(1)),(WT0(77),WB0(1)),(WT0(153),WC0(1)),   00025080
```

```
     * (WTO(229),WDO(1)),(WT1(1),WA1(1)),(WT1(77),WB1(1)),            00025090
     * (WT1(153),WC1(1)),(WT1(229),WD1(1))                            00025100
       EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                            00025110
C-----E=DEXP(.2D0), ER=1.0D0/E                                        00025120
       DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/    00025130
C--JO-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WTO ARRAY)        00025140
       DATA WAO/                                                      00025150
     *  2.1969101E-11, 4.1201161E-09,-6.1322980E-09, 7.2479291E-09,  00025160
     *-7.9821627E-09, 8.5778983E-09,-9.1157294E-09, 9.6615250E-09,   00025170
     *-1.0207546E-08, 1.0796633E-08,-1.1393033E-08, 1.2049873E-08,   00025180
     *-1.2708789E-08, 1.3446466E-08,-1.4174300E-08, 1.5005577E-08,   00025190
     *-1.5807160E-08, 1.6747136E-08,-1.7625961E-08, 1.8693427E-08,   00025200
     *-1.9650840E-08, 2.0869789E-08,-2.1903555E-08, 2.3305308E-08,   00025210
     *-2.4407377E-08, 2.6033678E-08,-2.7186773E-08, 2.9094334E-08,   00025220
     *-3.0266804E-08, 3.2534013E-08,-3.3672072E-08, 3.6408936E-08,   00025230
     *-3.7425022E-08, 4.0787921E-08,-4.1543242E-08, 4.5756842E-08,   00025240
     *-4.6035233E-08, 5.1425075E-08,-5.0893896E-08, 5.7934897E-08,   00025250
     *-5.6086570E-08, 6.5475248E-08,-6.1539913E-08, 7.4301996E-08,   00025260
     *-6.7117043E-08, 8.4767837E-08,-7.2583120E-08, 9.7366568E-08,   00025270
     *-7.7553611E-08, 1.1279873E-07,-8.1416723E-08, 1.3206914E-07,   00025280
     *-8.3217217E-08, 1.5663185E-07,-8.1482581E-08, 1.8860593E-07,   00025290
     *-7.3963141E-08, 2.3109673E-07,-5.7243707E-08, 2.8867452E-07,   00025300
     *-2.6163525E-08, 3.6808773E-07, 2.7049871E-08, 4.7932617E-07,   00025310
     *  1.1407365E-07, 6.3720626E-07, 2.5241961E-07, 8.6373487E-07,  00025320
     *  4.6831433E-07, 1.1916346E-06, 8.0099716E-07, 1.6696015E-06,  00025330
     *  1.3091334E-06, 2.3701475E-06, 2.0803829E-06, 3.4012978E-06/  00025340
       DATA WBO/                                                      00025350
     *  3.2456774E-06, 4.9240402E-06, 5.0005198E-06, 7.1783540E-06,  00025360
     *  7.6367633E-06, 1.0522038E-05, 1.1590021E-05, 1.5488635E-05,  00025370
     *  1.7510398E-05, 2.2873836E-05, 2.6368006E-05, 3.3864387E-05,  00025380
     *  3.9610390E-05, 5.0230379E-05, 5.9397373E-05, 7.4612122E-05,  00025390
     *  8.8951409E-05, 1.1094809E-04, 1.3308026E-04, 1.6511335E-04,  00025400
     *  1.9895671E-04, 2.4587195E-04, 2.9728181E-04, 3.6629770E-04,  00025410
     *  4.4402013E-04, 5.4589361E-04, 6.6298832E-04, 8.1375348E-04,  00025420
     *  9.8971624E-04, 1.2132772E-03, 1.4772052E-03, 1.8092022E-03,  00025430
     *  2.2045122E-03, 2.6980811E-03, 3.2895354E-03, 4.0238764E-03,  00025440
     *  4.9080203E-03, 6.0010999E-03, 7.3216878E-03, 8.9489225E-03,  00025450
     *  1.0919448E-02, 1.3340696E-02, 1.6276399E-02, 1.9873311E-02,  00025460
     *  2.4233627E-02, 2.9555699E-02, 3.5990069E-02, 4.3791529E-02,  00025470
     *  5.3150319E-02, 6.4341372E-02, 7.7506720E-02, 9.2749987E-02,  00025480
     *  1.0980561E-01, 1.2791555E-01, 1.4525830E-01, 1.5820085E-01,  00025490
     *  1.6058576E-01, 1.4196085E-01, 8.9781222E-02,-1.0238278E-02,  00025500
     *-1.5083434E-01,-2.9059573E-01,-2.9105437E-01,-3.7973244E-02,   00025510
     *  3.8273717E-01, 2.2014118E-01,-4.7342635E-01, 1.9331133E-01,  00025520
     *  5.3839527E-02,-1.1909845E-01, 9.9317051E-02,-6.6152628E-02,  00025530
     *  4.0703241E-02,-2.4358316E-02, 1.4476533E-02,-8.6198067E-03/  00025540
       DATA WCO/                                                      00025550
     *  5.1597053E-03,-3.1074602E-03, 1.8822342E-03,-1.1456545E-03,  00025560
     *  7.0004347E-04,-4.2904226E-04, 2.6354444E-04,-1.6215439E-04,  00025570
     *  9.9891279E-05,-6.1589037E-05, 3.7996921E-05,-2.3452250E-05,  00025580
     *  1.4479572E-05,-8.9417427E-06, 5.5227518E-06,-3.4114252E-06,  00025590
     *  2.1074101E-06,-1.3019229E-06, 8.0433617E-07,-4.9693681E-07,  00025600
     *  3.0702417E-07,-1.8969219E-07, 1.1720069E-07,-7.2412496E-08,  00025610
     *  4.4740283E-08,-2.7643004E-08, 1.7079403E-08,-1.0552634E-08,  00025620
     *  6.5200311E-09,-4.0284597E-09, 2.4890232E-09,-1.5378695E-09,  00025630
```

```
     * 9.5019040E-10,-5.8708696E-10, 3.6273937E-10,-2.2412348E-10,     00025640
     * 1.3847792E-10,-8.5560821E-11, 5.2865474E-11,-3.2664392E-11,     00025650
     * 2.0182948E-11,-1.2470979E-11, 7.7057678E-12,-4.7611713E-12,     00025660
     * 2.9415274E-12,-1.8170081E-12, 1.1221034E-12,-6.9271067E-13,     00025670
     * 4.2739744E-13,-2.6344388E-13, 1.6197105E-13,-9.9147443E-14,     00025680
     * 6.0487998E-14,-3.6973097E-14, 2.2817964E-14,-1.4315547E-14,     00025690
     * 9.1574735E-15,-5.9567236E-15, 3.9209969E-15,-2.5911739E-15,     00025700
     * 1.6406939E-15,-8.8248590E-16, 3.0195409E-16, 2.2622634E-17,     00025710
     *-8.0942556E-17,-3.7172363E-17, 1.9299542E-16,-3.3388160E-16,     00025720
     * 4.6174116E-16,-5.8627358E-16, 7.2227767E-16,-8.7972941E-16,     00025730
     * 1.0211793E-15,-1.0940039E-15, 1.0789555E-15,-9.7089714E-16/     00025740
       DATA WDO/                                                       00025750
     * 7.4110927E-16,-4.1700094E-16, 8.5977184E-17, 1.3396469E-16,     00025760
     *-1.7838410E-16, 4.8975421E-17, 1.9398153E-16,-5.0046989E-16,     00025770
     * 8.3280985E-16,-1.1544640E-15, 1.4401527E-15,-1.6637066E-15,     00025780
     * 1.7777129E-15,-1.7322187E-15, 1.5247247E-15,-1.1771155E-15,     00025790
     * 6.9747910E-16,-1.2088956E-16,-4.8382957E-16, 1.0408292E-15,     00025800
     *-1.5220450E-15, 1.9541597E-15,-2.4107448E-15, 2.9241438E-15,     00025810
     *-3.5176475E-15, 4.2276125E-15,-5.0977851E-15, 6.1428456E-15,     00025820
     *-7.3949962E-15, 8.8597601E-15,-1.0515959E-14, 1.2264584E-14,     00025830
     *-1.3949870E-14, 1.5332490E-14,-1.6146782E-14, 1.6084121E-14,     00025840
     *-1.4962523E-14, 1.2794804E-14,-9.9286701E-15, 6.8825809E-15,     00025850
     *-4.0056107E-15, 1.5965079E-15,-7.2732961E-18,-4.0433218E-16,     00025860
     *-6.5679655E-16, 3.3011866E-15,-7.3545910E-15, 1.2394851E-14,     00025870
     *-1.7947697E-14, 2.3774303E-14,-3.0279168E-14, 3.9252831E-14,     00025880
     *-5.5510504E-14, 9.0505371E-14,-1.7064873E-13/                    00025890
C--END OF JO FILTER WEIGHTS                                            00025900
C                                                                      00025910
C--J1-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT1 ARRAY)         00025920
       DATA WA1/                                                       00025930
     *-4.2129715E-16, 5.3667031E-15,-7.1183962E-15, 8.9478500E-15,     00025940
     *-1.0767891E-14, 1.2362265E-14,-1.3371129E-14, 1.3284178E-14,     00025950
     *-1.1714302E-14, 8.4134738E-15,-3.7726725E-15,-1.4263879E-15,     00025960
     * 6.1279163E-15,-9.1102765E-15, 9.9696405E-15,-9.3649955E-15,     00025970
     * 8.6009018E-15,-8.9749846E-15, 1.1153987E-14,-1.4914821E-14,     00025980
     * 1.9314024E-14,-2.3172388E-14, 2.5605477E-14,-2.6217555E-14,     00025990
     * 2.5057768E-14,-2.2485539E-14, 1.9022752E-14,-1.5198084E-14,     00026000
     * 1.1422464E-14,-7.9323958E-15, 4.8421406E-15,-2.1875032E-15,     00026010
     *-3.2177842E-17, 1.8637565E-15,-3.3683643E-15, 4.6132219E-15,     00026020
     *-5.6209538E-15, 6.4192841E-15,-6.8959928E-15, 6.9895792E-15,     00026030
     *-6.5355935E-15, 5.6125163E-15,-4.1453931E-15, 2.6358827E-15,     00026040
     *-9.5104370E-16, 1.4600474E-16, 5.6166519E-16, 8.2899246E-17,     00026050
     * 5.0032100E-16, 4.3752205E-16, 2.1052293E-15,-9.5451973E-16,     00026060
     * 6.4004437E-15,-2.1926177E-15, 1.1651003E-14, 5.8415433E-16,     00026070
     * 1.8044664E-14, 1.0755745E-14, 3.0159022E-14, 3.3506138E-14,     00026080
     * 5.8709354E-14, 8.1475200E-14, 1.2530006E-13, 1.8519112E-13,     00026090
     * 2.7641786E-13, 4.1330823E-13, 6.1506209E-13, 9.1921659E-13,     00026100
     * 1.3698462E-12, 2.0447427E-12, 3.0494477E-12, 4.5501001E-12,     00026110
     * 6.7870250E-12, 1.0126237E-11, 1.5104976E-11, 2.2536053E-11/     00026120
       DATA WB1/                                                       00026130
     * 3.3617368E-11, 5.0153839E-11, 7.4818173E-11, 1.1161804E-10,     00026140
     * 1.6651222E-10, 2.4840923E-10, 3.7058109E-10, 5.5284353E-10,     00026150
     * 8.2474468E-10, 1.2303750E-09, 1.8355034E-09, 2.7382502E-09,     00026160
     * 4.0849867E-09, 6.0940898E-09, 9.0913020E-09, 1.3562651E-08,     00026170
     * 2.0233058E-08, 3.0184244E-08, 4.5029477E-08, 6.7176304E-08,     00026180
```

```
      *  1.0021488E-07,  1.4950371E-07,  2.2303208E-07,  3.3272689E-07,     00026190
      *  4.9636623E-07,  7.4049804E-07,  1.1046805E-06,  1.6480103E-06,     00026200
      *  2.4585014E-06,  3.6677163E-06,  5.4714550E-06,  8.1626422E-06,     00026210
      *  1.2176782E-05,  1.8166179E-05,  2.7099223E-05,  4.0428804E-05,     00026220
      *  6.0307294E-05,  8.9971508E-05,  1.3420195E-04,  2.0021123E-04,     00026230
      *  2.9860417E-04,  4.4545291E-04,  6.6423156E-04,  9.9073275E-04,     00026240
      *  1.4767050E-03,  2.2016806E-03,  3.2788147E-03,  4.8837292E-03,     00026250
      *  7.2596811E-03,  1.0788355E-02,  1.5973323E-02,  2.3612041E-02,     00026260
      *  3.4655327E-02,  5.0608141E-02,  7.2827752E-02,  1.0337889E-01,     00026270
      *  1.4207357E-01,  1.8821315E-01,  2.2996815E-01,  2.5088500E-01,     00026280
      *  2.0334626E-01,  6.0665451E-02,-2.0275683E-01,-3.5772336E-01,     00026290
      *-1.8280529E-01,  4.7014634E-01,  7.2991233E-03,-3.0614594E-01,     00026300
      *  2.4781735E-01,-1.1149185E-01,  2.5985386E-02,  1.0850279E-02,     00026310
      *-2.2830217E-02,  2.4644647E-02,-2.2895284E-02,  2.0197032E-02/     00026320
       DATA #C1/                                                            00026330
      *-1.7488968E-02,  1.5057670E-02,-1.2953923E-02,  1.1153254E-02,     00026340
      *-9.6138436E-03,  8.2952090E-03,-7.1628361E-03,  6.1882910E-03,     00026350
      *-5.3482055E-03,  4.6232056E-03,-3.9970542E-03,  3.4560118E-03,     00026360
      *-2.9883670E-03,  2.5840861E-03,-2.2345428E-03,  1.9323046E-03,     00026370
      *-1.6709583E-03,  1.4449655E-03,-1.2495408E-03,  1.0805480E-03,     00026380
      *-9.3441130E-04,  8.0803899E-04,-6.9875784E-04,  6.0425624E-04,     00026390
      *-5.2253532E-04,  4.5186652E-04,-3.9075515E-04,  3.3790861E-04,     00026400
      *-2.9220916E-04,  2.5269019E-04,-2.1851585E-04,  1.8896332E-04,     00026410
      *-1.6340753E-04,  1.4130796E-04,-1.2219719E-04,  1.0567099E-04,     00026420
      *-9.1379828E-05,  7.9021432E-05,-6.8334412E-05,  5.9092726E-05,     00026430
      *-5.1100905E-05,  4.4189914E-05,-3.8213580E-05,  3.3045496E-05,     00026440
      *-2.8576356E-05,  2.4711631E-05,-2.1369580E-05,  1.8479514E-05,     00026450
      *-1.5980307E-05,  1.3819097E-05,-1.1950174E-05,  1.0334008E-05,     00026460
      *-8.9364160E-06,  7.7278366E-06,-6.6827083E-06,  5.7789251E-06,     00026470
      *-4.9973715E-06,  4.3215167E-06,-3.7370660E-06,  3.2316575E-06,     00026480
      *-2.7946015E-06,  2.4166539E-06,-2.0898207E-06,  1.8071890E-06,     00026490
      *-1.5627811E-06,  1.3514274E-06,-1.1686576E-06,  1.0106059E-06,     00026500
      *-8.7392952E-07,  7.5573750E-07,-6.5353002E-07,  5.6514528E-07,     00026510
      *-4.8871388E-07,  4.2261921E-07,-3.6546333E-07,  3.1603732E-07/     00026520
       DATA #D1/                                                            00026530
      *-2.7329579E-07,  2.3633470E-07,-2.0437231E-07,  1.7673258E-07,     00026540
      *-1.5283091E-07,  1.3216174E-07,-1.1420792E-07,  9.8831386E-08,     00026550
      *-8.5465227E-08,  7.3906734E-08,-6.3911437E-08,  5.5267923E-08,     00026560
      *-4.7793376E-08,  4.1329702E-08,-3.5740189E-08,  3.0906612E-08,     00026570
      *-2.6726739E-08,  2.3112160E-08,-1.9986424E-08,  1.7283419E-08,     00026580
      *-1.4945974E-08,  1.2924650E-08,-1.1176694E-08,  9.6651347E-09,     00026590
      *-8.3580023E-09,  7.2276490E-09,-6.2501673E-09,  5.4048822E-09,     00026600
      *-4.6739154E-09,  4.0418061E-09,-3.4951847E-09,  3.0224895E-09,     00026610
      *-2.6137226E-09,  2.2602382E-09,-1.9545596E-09,  1.6902214E-09,     00026620
      *-1.4616324E-09,  1.2639577E-09,-1.0930164E-09,  9.4519327E-10,     00026630
      *-8.1736202E-10,  7.0681930E-10,-6.1122713E-10,  5.2856342E-10,     00026640
      *-4.5707937E-10,  3.9526267E-10,-3.4180569E-10,  2.9557785E-10,     00026650
      *-2.5560176E-10,  2.2103233E-10,-1.9113891E-10,  1.6528994E-10,     00026660
      *-1.4294012E-10,  1.2361991E-10,-8.2740936E-11/                     00026670
C--END OF J1 FILTER WEIGHTS                                                 00026680
C                                                                           00026690
       NONE=0                                                               00026700
       IF(NEW.EQ.0) GO TO 100                                              00026710
       NSAVE=0                                                             00026720
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B                     00026730
```

```
      Y1=0.7358852661479794460D0/DBLE(B)                          00026740
  100 ZHANKS=(0.0,0.0)                                            00026750
      CMAX=(0.0,0.0)                                              00026760
      NF=0                                                        00026770
      Y=Y1                                                        00026780
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 131 (EITHER NEW=1 OR 0)  00026790
      ASSIGN 110 TO M                                             00026800
      I=131                                                       00026810
      Y=Y*E                                                       00026820
      GO TO 200                                                   00026830
  110 TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                            00026840
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                            00026850
      I=I+1                                                       00026860
      Y=Y*E                                                       00026870
      IF(I.LE.149) GO TO 200                                      00026880
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) NONE=1                00026890
C-----ESTABLISH TRUNCATION CRITERION (CMAX=CMPLX(TMAX(1),TMAX(2))  00026900
      CMAX=TOL*CMAX                                               00026910
      ASSIGN 120 TO M                                             00026920
      GO TO 200                                                   00026930
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END                    00026940
  120 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130 00026950
      I=I+1                                                       00026960
      Y=Y*E                                                       00026970
      IF(I.LE.283) GO TO 200                                      00026980
  130 Y=Y1                                                        00026990
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 130            00027000
      ASSIGN 140 TO M                                             00027010
      I=130                                                       00027020
      GO TO 200                                                   00027030
C-----CHECK FOR FILTER TRUNCATION AT LEFT END                     00027040
  140 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2).AND.       00027050
     * NONE.EQ.0) GO TO 190                                       00027060
      I=I-1                                                       00027070
      Y=Y*ER                                                      00027080
      IF(I.GT.0) GO TO 200                                        00027090
C-----RETURN WITH ISAVE=1 PRESET FOR POSSIBLE NEW=0 USE.          00027100
  190 ISAVE=1                                                     00027110
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE      00027120
      ZHANKS=ZHANKS/B                                             00027130
      RETURN                                                      00027140
C-----SAVE/RETRIEVE PSEUDO-SUBROUTINE (CALL FUN ONLY WHEN NECESSARY) 00027150
  200 G=SNGL(Y)                                                   00027160
      IF(NEW) 300,210,300                                         00027170
  210 IF(ISAVE.GT.NSAVE) GO TO 300                                00027180
      ISAVE0=ISAVE                                                00027190
  220 IF(G.EQ.GSAVE(ISAVE)) GO TO 240                             00027200
      ISAVE=ISAVE+1                                               00027210
      IF(ISAVE.LE.NSAVE) GO TO 220                                00027220
      ISAVE=ISAVE0                                                00027230
C-----G NOT IN COMMON/SAVE/----- EVALUATE FUN.                    00027240
      GO TO 300                                                   00027250
C-----G FOUND IN COMMON/SAVE/----- USE FSAVE AS GIVEN.            00027260
  240 C=FSAVE(ISAVE)                                              00027270
      ISAVE=ISAVE+1                                               00027280
```

```
C-----SWITCH ON ORDER N                                          00027290
  250 IF(N) 270,260,270                                          00027300
  260 C=C*WT0(I)                                                 00027310
      GO TO 280                                                  00027320
  270 C=C*WT1(I)                                                 00027330
  280 ZHANKS=ZHANKS+C                                            00027340
      GO TO M,(110,120,140)                                      00027350
C-----DIRECT FUN EVALUATION (AND ADD TO END OF COMMON/SAVE/)     00027360
  300 NSAVE=NSAVE+1                                              00027370
      C=FUN(G)                                                   00027380
      NF=NF+1                                                    00027390
      FSAVE(NSAVE)=C                                             00027400
      GSAVE(NSAVE)=G                                             00027410
      GO TO 250                                                  00027420
      END                                                        00027430
      REAL FUNCTION ASINH(X)                                     00027440
C--INVERSE HYPERBOLIC SIN FUNCTION                               00027450
C                                                                00027460
      REAL*8 X2                                                  00027470
      X2=X                                                       00027480
      ASINH=DLOG(X2+DSQRT(X2*X2+1.0D0))                          00027490
      RETURN                                                     00027500
      END                                                        00027510
      FUNCTION ERF(X)                                            00027520
C                                                                00027530
C   ERF COMPUTES THE ERROR FUNCTION TO ABOUT 7-PLACES.           00027540
C   SEE MATH. OF COMP., V.22,N.101,JAN,1968.                     00027550
C    ALSO, SEE ERFINV(X).                                        00027560
C                                                                00027570
      DIMENSION A1(19),A2(19)                                    00027580
      DATA A1/.70322500,.33050152,.20133975,.10863025,           00027590
     1 .46775523E-1,.15398573E-1,.38015077E-2,.69718379E-3,      00027600
     2 .94490927E-4,.94328117E-5,.69192752E-6,.37225234E-7,      00027610
     3 .14666061E-8,.42261614E-10,.88978652E-12,.13676044E-13,   00027620
     4 .15334234E-15,.12536751E-17,.74517E-20/                   00027630
      DATA A2/.24725517,.14422723,.86989455E-1,.43977338E-1,     00027640
     1 .17243963E-1,.50790696E-2,.11086065E-2,.17822802E-3,      00027650
     2 .21040458E-4,.18206632E-5,.11533099E-6,.53427503E-8,      00027660
     3 .18084859E-9,.44696823E-11,.80606884E-13,.10601364E-14,   00027670
     4 .10164928E-16,.710005E-19,0.0/                            00027680
      IF(X.EQ.0.0) THEN                                          00027690
        ERF=0.0                                                  00027700
        RETURN                                                   00027710
      ENDIF                                                      00027720
      B=2.*X/5.                                                  00027730
      S=SIN(B)                                                   00027740
      C=COS(B)                                                   00027750
      C2=C+C                                                     00027760
      ALP=C2*C-1.                                                00027770
      SUM=0.0                                                    00027780
      DO 10 N=1,19                                               00027790
        SUM=SUM+(A1(N)+C2*A2(N))*ALP**(N-1)                      00027800
10      CONTINUE                                                 00027810
      ERF=B/3.1415927+S*SUM                                      00027820
      RETURN                                                     00027830
```

```
      END                                                         00027840
      FUNCTION ERFINV(Y)                                          00027850
C                                                                 00027860
C  ERFINV COMPUTES THE INVERSE ERROR FUNCTION TO ABOUT 7-PLACES.  00027870
C  SEE MATH. OF COMP., V.22,N.101,JAN,1968.                       00027880
C  ALSO, SEE ERF(X).                                              00027890
C                                                                 00027900
      CHARACTER*16 XX                                             00027910
      DIMENSION T3(1:38),T4(0:26),T5(0:37),T6(0:25)               00027920
      DATA T3/.12046752,.16078199E-1,.26867044E-2,.49963473E-3,  00027930
     1 .98898219E-4,.20391813E-4,.43272716E-5,.93808141E-6,      00027940
     2 .20673472E-6,.46159699E-7,.10416680E-7,.23715100E-8,      00027950
     3 .54392841E-9,.12554899E-9,.29138180E-10,.67949422E-11,    00027960
     4 .15912343E-11,.37402505E-12,.88208776E-13,.20865090E-13,  00027970
     5 .49488041E-14,.11766395E-14,.28038557E-15,.66950664E-16,  00027980
     6 .16016550E-16,.38382583E-17,.9212851E-18,.2214615E-18,    00027990
     7 .533091E-19,.128488E-19,.31006E-20,.7491E-21,.1812E-21,   00028000
     8 .439E-22,.106E-22,.26E-23,.6E-24,.2E-24/                  00028010
      DATA T4/.91215880,-.16266202E-1,.43355647E-3,.21443857E-3, 00028020
     1 .26257511E-5,-.30210911E-5,-.12406061E-7,.62406609E-7,    00028030
     2 -.54012479E-9,.14232079E-8,.34384028E-10,.33584870E-10,   00028040
     3 -.14584289E-11,-.81021743E-12,.52532409E-13,.19711541E-13,00028050
     4 -.17494334E-14,-.48005966E-15,.55730299E-16,.11632605E-16,00028060
     5 -.17262489E-17,-.2784973E-18,.524481E-19,.65270E-20,      00028070
     6 -.15707E-20,-.1475E-21,.450E-22/                          00028080
      DATA T5/.95667971,-.23107004E-1,-.43742361E-2,-.57650342E-3,00028090
     1 -.10961022E-4,.25108547E-4,.10562336E-4,.27544123E-5,     00028100
     2 .43248450E-6,-.20530336E-7,-.43891537E-7,-.17684010E-7,   00028110
     3 -.39912890E-8,-.18693241E-9,.27292274E-9,.13281721E-9,    00028120
     4 .31834248E-10,.16700608E-11,-.20364650E-11,-.96484681E-12,00028130
     5 -.21956727E-12,-.95689813E-14,.13703257E-13,.62538505E-14,00028140
     6 .14584615E-14,.10781240E-15,-.70922999E-16,-.39141178E-16,00028150
     7 -.11165921E-16,-.15770366E-17,.2853149E-18,.2716662E-18,  00028160
     8 .957770E-19,.176835E-19,-.9828E-21,-.20464E-20,-.802E-21, 00028170
     9 -.1650E-21/                                               00028180
      DATA T6/.98857506,.10857705E-1,-.17511651E-2,.21196993E-4, 00028190
     1 .15664871E-4,-.51904169E-5,-.37135790E-7,.12174309E-8,    00028200
     2 -.17681155E-9,-.11937218E-10,.38025054E-12,-.66018832E-13,00028210
     3 -.87917055E-14,-.35068693E-15,-.69722150E-16,-.10956794E-16,00028220
     4 -.11536390E-17,-.1326235E-18,-.263938E-19,.5341E-21,      00028230
     5 -.2261E-20,.9552E-21,-.525E-21,.2487E-21,-.1134E-21,.42E-22/00028240
      X=Y                                                         00028250
      X1=ABS(X)                                                   00028260
      IF(X1.GE.1.0) THEN                                          00028270
        ENCODE(16,1,XX) X1                                        00028280
    1   FORMAT(E16.8)                                             00028290
        IF(X1.GT.1.000001)CALL ERRMSG('ABS(X)='//XX//            00028300
     1   ' >1.000001 IN [ERFINV]',0,6,0)                          00028310
        CALL WARN('ABS(X)='//XX//                                00028320
     2   ' >=1.0 IN [ERFINV]; X=0.9999998*SIGN(1.,X) USED.',0,6,0,*2)00028330
    2   X=0.9999998*SIGN(1.,X)                                    00028340
      ENDIF                                                       00028350
      X1=1.-X                                                     00028360
      IF(X.GE.0.8.AND.X.LE.0.9975) THEN                           00028370
        BETA=SQRT(-ALOG(1.-X*X))                                  00028380
```

```
        R=0.0                                                          00028390
        DO 10 N=0,26                                                   00028400
10         R=R+T4(N)*TCHEB(N,-1.54801304*BETA+2.5654901)               00028410
        ERFINV=BETA*R                                                  00028420
      ELSE IF(X1.GE.5E-16.AND.X1.LE.25E-4) THEN                        00028430
        BETA=SQRT(-ALOG(1.-X*X))                                       00028440
        R=0.0                                                          00028450
        DO 20 N=0,37                                                   00028460
20         R=R+T5(N)*TCHEB(N,-.55945763*BETA+2.2879157)                00028470
        ERFINV=BETA*R                                                  00028480
      ELSE IF(X1.LT.5E-16) THEN                                        00028490
        BETA=SQRT(-ALOG(1.-X*X))                                       00028500
        SBETA=SQRT(BETA)                                               00028510
        R=0.0                                                          00028520
        DO 30 N=0,25                                                   00028530
30         R=R+T6(N)*TCHEB(N,-9.1999924/SBETA+2.7949908)               00028540
        ERFINV=BETA*R                                                  00028550
      ELSE                                                             00028560
        R=0.0                                                          00028570
        A=X*X/.32-1.                                                   00028580
        DO 40 N=1,38                                                   00028590
40         R=R+T3(N)*TCHEB(N,A)                                        00028600
        ERFINV=X*(.99288538+R)                                         00028610
      ENDIF                                                            00028620
      RETURN                                                           00028630
      END                                                              00028640
      INTEGER FUNCTION LOC(I,J)                                        00028650
C--GETS ACTUAL ADDR OF A(I,J)=A(J,I) SYMMETRIC MATRIX                  00028660
C  STORED AS THE VECTOR A(LOC(I,J)) OF N*(N+1)/2 ELEMENTS--            00028670
C  WHERE ANY I,J.LE.N MAY BE USED (N NOT EXPLICITLY NEEDED)...         00028680
C                                                                      00028690
      IF(I-J) 10,20,20                                                 00028700
   10 LOC=I+(J*J-J)/2                                                  00028710
      RETURN                                                           00028720
   20 LOC=J+(I*I-I)/2                                                  00028730
      RETURN                                                           00028740
      END                                                              00028750
      SUBROUTINE NL2SOL(N, P, X, CALCR, CALCJ, IV, V, UIPARM, URPARM,  00028760
     1                  UFPARM)                                        00028770
C** VAX-11/780 VERSION (12/18/80) MODIFIED BY                          00028780
C** W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.             00028790
```

$$$$$ Because of the length of NL2SOL and related subprograms, the rest
      of the listing has been suppressed; however, the complete code is
      available on the distributed tape.